Noname manuscript No.
(will be inserted by the editor)

Leveraging creativity in requirements elicitation within agile software development: a systematic literature review

Ainhoa Aldave · Juan M. Vara · David Granada · Esperanza Marcos

Received: date / Accepted: date

Abstract When applied to requirements elicitation in agile Software Development, the term "Creative thinking" suggests the need to fill the gap that Agile Methodologies have as regards the lack of attention that is paid to design and an insufficiency of innovation. This synergy between creativeness and agility has arisen as a new means of bringing innovation and flexibility to increasingly demanding software. The aim of the present study is to employ a systematic review to investigate the state-of-the-art of those approaches that empower creativity in requirements elicitation within Agile Software Development, in addition to the benefits, limitations and the strength of evidence of these approaches. The review was carried out by following the guidelines proposed by the Dr. Kitchenham. The search strategy identified 1451 studies, 17 of which were eventually classified as primary studies. The selected studies contained 13 different and unique proposals. These approaches provide evidence that empowering creativity in requirements elicitation can be successfully implemented in real software projects. We specifically observed that projects related to user interface development, such as those for mobile or web applications, are good candidates for the use of these approaches. We have also found that agile methodologies such as Scrum, Extreme Programming or methodologies based on rapid modelling are preferred when introducing creativity into requirements elicitation. Despite being a new research field, there is a mixture of techniques, tools and processes that have already been and are currently being successfully tested in industry. Finally, we have found that, although creativity is an important ingredient with which to bring about innovation, it is not always sufficient to generate new requirements and this needs to be accompanied by user engagement and a specific context in which proper conditions, such as flexibility, time or resources, have to be met.

Keywords Software Development · Software Project Management · Agile Methodologies · Requirements Elicitations · Creative Thinking · Systematic Review

1 Introduction

In today's software development environments, in which needs, technology, complexity and exigencies evolve so quickly, the need for innovation and agility has become crucial and very often the reason for the disruptive gap between success and failure [42,43]. Software ideas need to be functional as soon as possible and sufficiently original to beat the competition and meet the expectations of their ever more demanding users.

Agile methodologies such as Extreme Programming, Kanban or Scrum have, therefore, emerged to deal with the increasing complexity in software engineering and to handle the inevitable changes in requirements throughout the lifecycle of software.

Agile software development methodologies have already proven their efficacy as regards delivering software more quickly, improving customer collaboration, estimating time and handling defects in processes [22]. While agility is already established in organizations of all sorts, the time has come to confront its limitations.

A. Aldave, J.M. Vara, D. Granada, E. Marcos
Kybele Research Group, Rey Juan Carlos University,
Calle Tulipán S/N, 28933, Móstoles, Madrid, Spain

A. Aldave
E-mail: a.aparicioa@alumnos.urjc.es

J.M. Vara
E-mail: juanmanuel.vara@urjc.es

D. Granada
E-mail: david.granada@urjc.es

E. Marcos
E-mail: esperanza.marcos@urjc.es

Some studies suggest that the existing agile approaches employed during the requirements analysis phase tend to focus solely on scoping and simplicity rather than on problem solving and discovery [85,32]. This results in the inability to develop innovative solutions.

Additionally, even although it is claimed that agile methodologies provide collaboration between the customer and the development team, very few define how the people on the team playing the role of the customer can learn what the real end user needs and how they can accurately represent those needs [S17].

If agility is apparently not a sufficient means to provide innovative solutions, how can we enhance innovation into software development?

Innovation has been defined as the successful implementation of creative ideas [27], which leads us to the question: how can we successfully enhance creativity in software development?

Within the field of Design, creativity has been widely considered as an essential prerequisite to enable the appearance of innovative ideas in product development [77]. Some creative approaches, such as Design Thinking, have suggested models and strategies whose objective is to facilitate the delivery of innovative ideas [45,8,54].

But, why introduce creativity into only Design? Would it be possible to apply similar approaches into other phases of software development?

Software development usually starts with requirements analysis, which plays a determining role in the development process and is still considered as one of the most critical activities in any software development project [18].

In fact, we have discovered that, during the last decade, engineers have intensively started to underline the importance of creative thinking in requirements elicitation (RE) as a decisive factor for building competitive and innovative products [48,49].

Several systematic reviews on the application of agile methodologies in the requirements engineering currently exist, each of them focused on different aspects of the process of defining, documenting and maintaining requirements [17, 33,70,26,68].

We have also found some works studying the state-of the art as regards creativity in agile systems development [15] or creativity in requirements engineering [44]. However, we have not, to date been able to find, a systematic review focused on combining both, creative approaches for requirements elicitation within agile software development. This paper aims to fill this gap by presenting a systematic review of the existing work in this field.

The main goal of this paper is to study the state-of the art of the approaches that strengthen creativity in requirements elicitation within agile software development.

The following sub-goals have, therefore, been defined:

- The research on background studies: requirements elicitation, agile methodologies and creativity.
- The definition of a criterion that will serve to select and evaluate relevant studies.
- Teh execution of a systematic literature review.
- The review and summary of the selected studies and the identification of trends.

The remainder the paper is organized as follows: Section 2 describes the background that was studied prior to carring out this work: requirements elicitation, agile methodologies and creativity; Section 3 explains the method adopted to conduct the systematic review; Section 4 provides a detailed review of each of the results of the systematic review; Section 5 summarizes the findings and provides a discussion, and finally, Section 6 summarizes the main conclusions derived from this paper and outlines our future work.

2 Research context

Before exploring the contents of this paper, we consider that it is necessary to provide precise definitions of some of the terms used throughout this work, which are principally related to the Requrements Elicitation, Agile Methodologies and Creativity.

2.1 Requirements Elicitation

Requirements elicitation is the process of defining stakeholders' needs and gathering this information together in an understandable manner such that developers can construct a system that will address those needs [84].

Although requirements elicitation is already a relatively mature area in the software development industry [29] it is still considered to be one of the most critical activities in any software development project [18].

One of the main challenges of requirements elicitation is that of fully understanding what the users really want. Interaction with the users usually takes place via natural language, which is not always straightforward. The consequence is that users tend to provide incomplete or ambiguous requirements. Another problem is that the requirements gathered during the early stages of the project are likely to evolve or be discarded in later phases of the project. A lack of user involvement or having unrealistic expectations are also common problems. Since the social context is much more crucial than the technical one, these issues cannot be solved in a purely technological way [25].

Some of the typical activities in the requirements elicitation process include [87]:

- Understanding the application domain in which the system will operate.

- Identifying the sources of requirements: stakeholders, subject matter experts, existing systems and documentation etc.
- Analyzing the stakeholders, anyone who has an interest in the system or is affected by the development and implementation of the system.
- Selecting the techniques, approaches and tools tha will be used to elicit the requirements from stakeholders and other sources.

Although a combination of techniques is usually employed during the different stages of the software development project, requirements analysts tend to choose a technique that they already know, have worked with in a previous project, is that prescribed by a concrete methodology being used or that they sense will be effective in the current context [28].

Requiremenst elicitation techniques have been studied in depth in a wide number of studies [84,18,28,25,87,53]. Since the number of different techniques and approaches is extensive, there is no standard categorization. Some of the most popular include: interviews, questionnaires, surveys, scenarios, brainstorming, user stories, workshops, role playing and prototyping.

Within the context of this paper, we focus principally on requirements elicitation techniques and approaches that promote creativity and user engagement.

2.2 Agile Methodologies

Within the field of software development, in which needs, technology, complexity and exigencies evolve so quickly [42, 34], the need for innovation and agility has become crucial and very often for the reason of the disruptive gap between success and failure. Agile methodologies, such as the Dynamic Software Development Method [78], Extreme Programming [6], Crystal [13], Feature-driven development [60], Kanban [3] or Scrum [74], have emerged to deal with the increasing complexity in software engineering and to handle the inevitable changes in requirements throughout their lifecycle [34].

But what does agile mean? The terms “agile” and “agility” can be traced back to the manufacturing industry in 1991 when “lean development” emerged in manufacturing with the aim of eliminating waste, amplifying learning, delivering as fast as possible and empowering teams [63,34]. The idea of iterative and incremental development used in most agile process models can, however, be found in the 1930s when a quality expert at Bell Labs used this practice to improve product quality [40]. The practice of using a prototype of working software as the primary measure of progress is one of the principles in agile modeling.

In 2001, a group of 17 independent-minded software consultants and practitioners gathered together and signed the Agile Software Development Manifesto, which promulgated the following core values and principles:

- Values:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- Principles:
  - The highest priority is to satisfy the customer through the early and continuous delivery of valuable software
  - Welcome changing requirements, even late thedevelopment. Agile processes harness change forthe customer’s competitive advantage
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale
  - Business people and developers must work together daily throughout the project
  - Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done
  - The most efficient and effective method by which to convey information to and within a development team is face-to-face conversation
  - Working software is the primary measure of progress
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
  - Paying continuous attention to technical excellence and good design enhances agility
  - Simplicity -the art of maximizing the amount of work not done- is essential
  - The best architectures, requirements and designs emerge from self-organizing teams
  - At regular intervals, the team should reflect on how to become more effective, then tune and adjust its behavior accordingly

Agile Methodologies have already proven beneficial as regards reducing the delivery time of working software, improving customer collaboration, estimating time and handling defect processes [22].

In the following subsections, we describe some of the most popular Agile Methodologies: Extreme Programming, Kanban and Scrum.

2.2.1 Extreme Programming

Extreme Programming [6], familiarly known as XP, originated in 1999 and is a style of software development that focuses on excellent application of programming techniques, clear communication, and teamwork to produce quality software at a sustainable pace. While XP began as a methodology addressing small teams working on internal projects, teams worldwide have used XP for shrink-wrap, embedded, and large-scale projects as well. XP additionally adapts to vague or rapidly changing requirements, but can also be successfully used in those cases in which requirements do not appear to be volatile, such as porting projects. Figure 1 provides a visual summary of the different techniques used in Extreme Programming.

XP is based on the following core values (`http://www.extremeprogramming.org`):

- *Simplicity*: we will do what is needed and asked for, but no more. This will maximize the value created for the investment made to date. We will take small simple steps to attain our goal and mitigate failures as they happen. We will create something we are proud of and maintain it long term for reasonable costs.
- *Communication*: everyone is part of the team and we communicate face to face daily. We will work together on everything from requirements to code. We will create the best solution to our problem that we can together.
- *Feedback*: we will take every iteration commitment seriously by delivering working software. We will demonstrate our software early and often then listen carefully and make any changes needed. We will talk about the project and adapt our process to it, not the other way around.
- *Respect*: everyone will give and feel the respect they deserve as a valued team member. Everyone will contribute value, even if it is simply enthusiasm. Developers will respect the expertise of the customers and vice versa. Management will respect our right to accept responsibility and receive authority over our own work.
- *Courage*: we will tell the truth about progress and estimates. We will not document excuses for failure because we plan to succeed. We will not fear anything because no one ever works alone. We will adapt to changes whenever they occur.

2.2.2 Kanban

In software development, Kanban is an agile methodology that places emphasis on continual delivery without overloading the team by limiting work in progress.

It is based on a system originally developed in 1980 by the industrial engineer Taiichi Ohne at Toyota to improve manufacturing efficiency [58,3]. This system consists of a few kanban (or cards) equivalent to the (agreed) capacity of a system, as shown in the example in Figure 2. One card is attached to one piece of work. Each card acts as a signaling mechanism. A new piece of work can be started only when a card is available. This free card is attached to a piece of work and follows it as it flows through the system. When there are no more free cards, no additional work can be started. Any new work must wait in a queue until a card becomes available. When some work is completed, its card is detached and recycled. With a card now free, a new piece of work in the queuing can be started.

In software development, each card represents a work item. These cards can be virtual or physic. Card walls have become a popular visual control mechanism in Agile software development. Using either a cork notice board with index cards pinned to a board, or a whiteboard with sticky notes to track work-in-progress has become commonplace.

Kanban as a methodology in software development arose at Corbis from 2006 through 2008 and has continued to evolve in the wider Lean software development community in the years since [3]. Today, it is used to limit a team's work-in-progress to a set capacity and to balance the demand on the team against the throughput of their delivered work. By doing this, we can achieve a sustainable pace of development, thus enabling all individuals to achieve a work versus personal life balance.

This system has been shown to improve customer satisfaction through regular, dependable, high-quality releases of valuable software. It has also been shown to improve productivity, quality, and lead times. In addition, there is evidence that Kanban is a pivotal catalyst for the emergence of a more agile organization through evolutionary cultural change.

2.2.3 Scrum

Scrum is an agile of iterative and incremental product delivery method that uses frequent feedback and collaborative decision making [76]. It is based on a 1986 paper written by Hirotaka Takeuchi and Ikujiro Nonaka for the Harvard Business Review titled "The New New Product Development Game" [83]. In this paper, the authors used the sport of rugby as a metaphor to describe the benefits of self-organizing teams in innovative product development and delivery. Jeff Sutherland, Ken Schwaber, and Mike Beedle took the ideas from this paper and applied it to their field of software development. They called their new method Scrum, after the rugby term that describes how teams form a circle and go for the ball to get it back into play again. They first applied this method at Easel Corporation in 1993. Schwaber and Beedle wrote about their experiences in their book *Agile Software Development with Scrum* in 2002 [73], which was followed by Schwaber's book *Agile Project Management with Scrum* in 2004 [72].

The core element of Scrum is a Sprint, a time-box of one month or less during which a "Done", useable and potentially releasable product Increment is created. A new Sprint

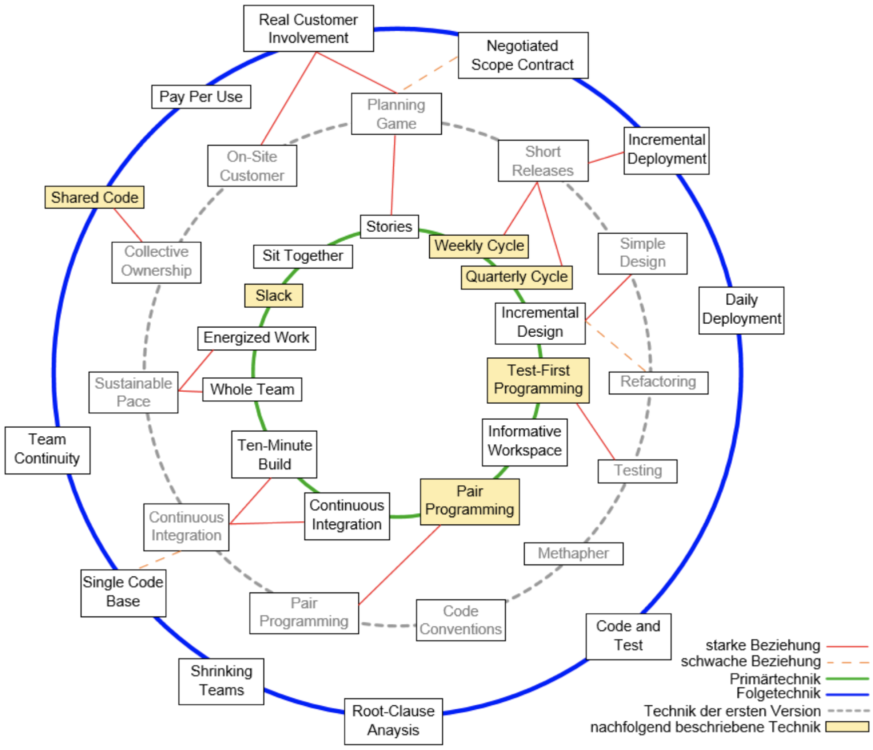


Fig. 1 XP Techniques (from [24])

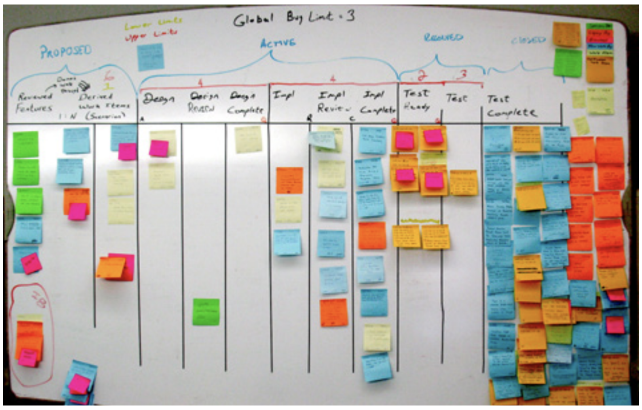


Fig. 2 Example of Kanban (from [3])

starts immediately after the conclusion of the previous Sprint [74]. Figure 3 shows the events, roles and artifacts of the scrum framework.

Scrum prescribes 4 formal events:

1. *Sprint Planning*: an event with a maximum duration of eight hours in which a one-month Sprint is planned. This plan is created by the collaborative work of the entire Scrum Team.
2. *Daily Scrums*: a 15-minute time-boxed event that allows the Development Team to synchronize activities and create a plan for the next 24 hours
3. *Sprint Review*: a four-hour revision meeting held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed
4. *Sprint Retrospective*: a three-hour meeting that occurs after the Sprint Review and prior to the next Sprint Planning. It is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

Scrum defines three types of roles: the ScrumMaster, the Product Owner and the Team.

1. The *Team* consists of professionals who do the work of delivering a potentially releasable Increment of “Done” product at the end of each Sprint. They are self-organizing and cross-functional.
2. The *ScrumMaster* is responsible for ensuring that Scrum is understood and enacted. Scrum Masters do this by ensuring that the Scrum Team adheres to Scrum theory, practices, and rules.
3. The *Product Owner* is responsible for maximizing the value of the product and the work of the Development Team and is the sole person responsible for managing the Product Backlog.

Scrum contains 3 types of artifacts that represent work or value to provide transparency and opportunities for inspection and adaptation: the Product Backlog, the Sprint Backlog and the Increment.

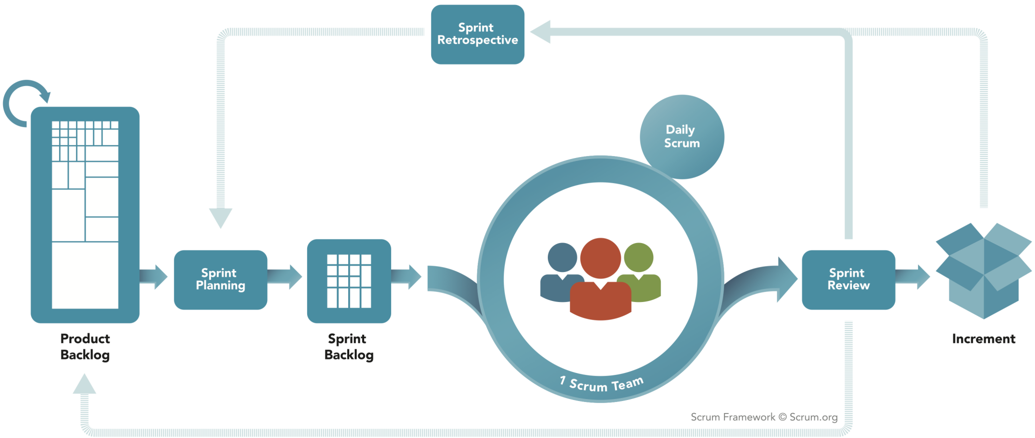


Fig. 3 Scrum framework (from https://www.scrum.org)

1. *Product Backlog*: an ordered list of everything that might be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.
2. *Sprint Backlog*: the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.
3. *Increment*: the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.

2.3 Creativity

The concept of creativity has been studied from many points of views for decades, and has its roots in the 1950s [66]. Creativity, as first mentioned by Barron [4] and Stein [79] requires both originality and effectiveness. Sternberg and Lubart [80] define creativity as “the ability to produce work that is both novel (i.e. original, unexpected) and appropriate (i.e. useful, adaptive concerning task constraints)”.

Unlike the thoughts that creativity belongs exclusively to a selected few with this ability, such as Beethoven, Mozart or Einstein, there are other theories, such as that proposed by Kaufman and Beghetto [36], that have distinguished among four levels of creativity: Big-C, that consists of clear-cut, eminent creative contributions found in works of creative genius; Little-c or everyday creative activities for non-experts; Mini-c, new and personally meaningful interpretations of experiences, actions and events; and Pro-c, found in professional level expertise in non-creative fields. This theory shows us that almost anyone with the right tools, experience or under certain circumstances could provide creative contributions.

Within the field of Design, creativity has been widely considered as an essential prerequisite by which to bring about innovative ideas [77], and some popular approaches, such as Design Thinking, have emphasized that the role of design in development has changed from having a mere focus on the aesthetic aspects and attractiveness of products to delivering innovative ideas in order to create those products [45,8,9,54]. An introduction to Design Thinking will be provided in Section 2.3.1.

With regard to Requirements Elicitation, during the last decade, engineers have intensively started to underline the importance of creative thinking in requirements elicitation as a decisive factor for the construction of competitive and innovative products [48,49]. We shall provide details on this in Section 2.3.2.

2.3.1 Creativity in Design - Design Thinking

Design as a way of thinking is an innovative approach that started in the late 60s [75] and expanded in the 80s and 90s [23,65]. Unlike analytical thinking, which is based on data sources and past knowledge to predict future needs, design thinking is proposed as a creative and non-lineal problem-solving approach.

The main foundations of design thinking are [62]:

- A focus on human behavior
- A lot of prototyping and testing
- Iteration and flexibility
- Collaboration and multidisciplinary teams

Above all, Design Thinking is a human centered approach focused on the users and their needs, which aims to ensure that the solution developed meets a real user need. It is also a collaborative way of working that allows development teams to make better decisions, quickly test ideas with the user, including feedback as a fundamental piece of the solution process. Since it deals with both the problem and the solution as something to be explored, the process requires diverging on many possible solutions (Empathy and

Ideate phases) and converging on a focused direction (Define, Prototype and Evaluate phases) [45,62,11]. Figure 4 shows the Design Thinking process.

The first phase, “Empathy”, consists of a set of activities aimed at understanding users’ needs by observing use cases or scenarios.

Immediately after the environment and the physical and emotional needs of the users have been understood, the information that has been collected is analyzed and summarized using tools such as personas and empathy maps. This phase of the “Define” process is responsible for bringing clarity and focus to the design space.

The next stage, “Ideate”, focuses on the generation of new ideas in order to avoid obvious solutions and thus increase the potential for innovation . Brainstorming, mind mapping or sketching are common techniques used in this phase.

The next phase, “Prototype”, consists of the iterative generation of artifacts (anything that a user can interact with) that support the elaboration and evaluation of product concepts with the goal of discovering which ideas proposed during the “Ideate” phase are right or wrong.

These prototypes are tested by the users during the “Evaluate” phase, which aims to show and learn from the users in order to develop better prototypes.

2.3.2 Creativity in Requirements Engineering

A number of works have been published in the field of creativity in Requirements Engineering during the last decade. While some works have focused on the theoretical side by presenting theoretical frameworks [57], or studying creativity theory in agile methods [15], others have focused on identifying which creativity techniques are important in requirements elicitation [67] or have studied the empirical evidence that exists as regards the use of creativity techniques during the requirements elicitation process [82].

Lemos et al. [44] carried out a mapping study on creativity in Requirements Engineering that confirms the relevance of treating requirements as a creative problem solving exercise. This study comes to the conclusion that authors are proposing new solutions to promote the use of creativity in the requirements elicitation process, in addition to evaluating these proposals in industrial case studies and experiments. The study also discovers that the majority of studies focus on the requirement elicitation phase. It was for this reason that we decided to focus on requirement elicitation rather than on requirements engineering in general.

Although we have found works that analyze agility within Requirements Engineering [70] or creativity in requirements engineering [44] we have not found a systematic review that makes an approach combining creativity in requirements elicitation specifically looking into agile software development.

3 Systematic literature review process

This section describes the process followed in order to carry out the systematic review presented in this paper. It explains the methodology, the search strategy and the procedures that have been carried out to select and extract information.

3.1 Methodology of the Systematic Literature Review

A systematic literature review [37,7] is a means of identifying, evaluating and interpreting all available research that is relevant to a particular research question, or topic area, or phenomenon of interest in a methodic and reproducible manner. Because of the large number of available sources and the hundreds of thousands of pieces of existing data, a systematic review must be carried out by following a rigorous method. Barbara Kitchenham proposed a method with which to perform a systematic literature review in software engineering that consists of a three-step process: planning, execution and result analysis [37].

However, this method is described at a relatively high level without considering the impact of question type on the review procedures, or providing a detailed specification of the mechanisms that are needed to undertake metaanalysis. It was for these reasons that Biolchini et al. [7] proposed a new process based on Kitchenham’s proposal to perform a systematic review in which they presented a new approximation composed of four stages: planning, execution, result analysis and packaging. This process is described in Figure 5.

The research objectives are listed and a review protocol defined during the planning phase. This protocol specifies the central research question and the methods that will be used to execute the review. The execution stage involves the identification of primary studies, along with selection and evaluation in accordance with the inclusion and exclusion criteria established in the review protocol. During the result analysis phase, data from the primary studies is extracted and summarized. Concurrently, during these phases, all the results obtained are stored. This storage is known as the packaging phase, which is performed throughout the whole process. Additionally, there are two checkpoints (represented by diamonds) that allow the detection of problems or errors after the completion of the planning and execution phases [7].

In order to follow this new proposal, Biolchini et al. provided Software Engineering researchers with a template that would serve as a guideline when conducting systematic reviews. Since this template is focused on systematic reviews based on empirical studies, and the objective of this paper is to carry out a literature analysis of a very new area in which proper empirical evidence might be very limited, we

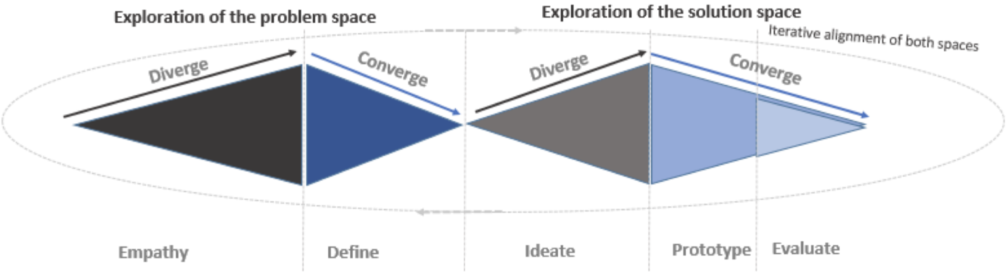

Exploration of the problem space
Exploration of the solution space
Iterative alignment of both spaces
Diverge
Converge
Diverge
Converge
Empathy
Define
Ideate
Prototype
Evaluate

Fig. 4 Design Thinking Process (adapted from [23])

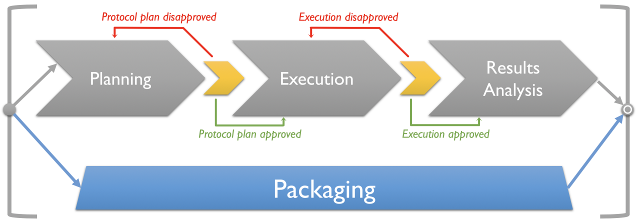


Fig. 5 Systematic Review Process (adapted from [7])

have followed this template but have excluded the following subsections: control, population, experimental design, results statistical calculus, sensitivity analysis and plotting.

3.2 Objective of the systematic review and definition of research questions

The aim of the present study is to employ a systematic review to investigate the state of the art of the techniques or methodologies that apply creative approaches to requirements elicitation in Agile Software Development, along with the benefits and limitations and the strength of evidence of these approaches.

Agile Methodologies have already proven beneficial as regards reducing the delivery time of working software, improving customer collaboration, estimating time and handling defects processes [22]. Some studies, however, suggest that the existing agile approaches, when used during the requirements analysis phase, are prone to focus only on scoping and simplicity rather than on problem solving and discovery [85,32]. Considering that requirements elicitation plays a determining role in the development process and it is characterized by ambiguity, uncertainty, and unpredictability [70], it has been proposed that agile processes be extended with creative techniques in order to manage complex software developments [31,49,47,44]. We have attempted to identify the state of the art in this topic by defining the following research questions (RQ):

- *RQ1*. What methodologies or techniques that involve creative approaches in requirements elicitation (RE) in agile software development (ASD) exist?
- *RQ2*. What empirical evidence of the application of creative techniques for RE in ASD exists?
- *RQ3*. What is it known about the benefits and limitations of using creative approaches for RE in ASD?
- *RQ4*. In what sort of organizations or projects using ASD could the use of these creative approaches be most suitable?
- *RQ5*. Which agile methodologies are being used to integrate creative techniques into agile software development RE processes?

3.3 Data sources and query strings

Once the research questions had been established, a set of keywords that matched the research objectives was selected. This set of keywords covered three main domains: “Creativity”, “Requirements” and “Agile”. Alternative spellings and synonyms were identified for each domain. The complete set of keywords can be seen in Table 1. The selected keywords were subsequently connected with Boolean operators to create a search query as follows: (“design thinking” OR creative OR creativity) AND (requirements OR “product backlog” OR preconditions OR specifications OR modeling OR analysis OR “user stories”) AND (agile OR scrum OR Kanban OR lean OR crystal OR xp OR “extreme programming”).

The objective was to create a search query that would cover at least one term of each domain. This search query was then adapted to the syntax of the different search engines

Table 1 Set of keywords

Requirements	Creativity	Agile
Requirement	Creativity	Agile
Product Backlog	Creative	Scrum
Precondition		Kanban
Specification		Lean
Modeling		Crystal
Analysis		XP
User Stories		Extreme Programming

A fourth block of keywords was initially included in the query related to “Information Systems and Software Engineering”. However, as evidenced by the search results, it was later observed that this block was overcomplicating the search query without providing additional relevant results. Something similar occurred while testing the extra search query terms, such as other types of agile methodologies (feature-driven development, Dynamic systems development method) or synonyms such as “agility”. That said, owing to the limitation of terms in some search engines or the inability to attain additional relevant results with those synonyms, it was decided to keep the query as simple as possible.

During this stage, data sources were identified and a manual search process was executed in specific electronic databases. The results obtained were analyzed and the data sources were sorted. We initially considered a set of digital libraries that was considerably reduced, because after obtaining the results some digital libraries did not provide us with relevant information. The final list of data sources that has been employed for this systematic review is shown in Table 2.

Table 2 Set of selected data sources

Source	Website
ACM	http://portal.acm.org
Google Scholar	http://scholar.google.com
IEEE Xplore	http://ieeexplore.ieee.org
ISI Web of Science	http://www.isiknowledge.com/
Science Direct	http://www.sciencedirect.com
Scopus	https://www.scopus.com

3.4 Study selection

In order to proceed with the primary study selection, this systematic review followed the procedure proposed by Pino, García and Piattini [61], which is based on the proposal of Biolchini et al. [7]. This procedure is described in Figure 6.

The search query described in the previous section was adapted to each of the selected sources by considering the following criteria:

1. Restriction by Title &amp; Abstract: if the data source allowed us to restrict the search query only to the title and abstract or title, abstract and keywords, the query was modified accordingly. For example, in the ACM Digital Library we added the prefix *acmdlTitle*: to search for a keyword only within the title of a study and the prefix *recordAbstract*: to search only in the abstract section. Similarly, in ScienceDirect and Scopus we included the prefix *TITLE-ABS-KEY* before each keyword. This is an example of an adapted search query:

```
( TITLE-ABS-KEY (“design thinking”) OR TITLE-ABS-KEY ( creative) OR
TITLE-ABS-KEY ( creativity ) ) AND ( TITLE-ABS-KEY ( requirements )
OR TITLE-ABS-KEY (“product backlog”) OR TITLE-ABS-KEY (
preconditions) OR TITLE-ABS-KEY ( specifications) OR TITLE-ABS-KEY
( modeling ) OR TITLE-ABS-KEY ( analysis ) OR TITLE-ABS-KEY (“user
stories”)) AND ( TITLE-ABS-KEY ( agile ) OR TITLE-ABS-KEY ( scrum )
OR TITLE-ABS-KEY ( kanban ) OR TITLE-ABS-KEY ( lean ) OR
TITLE-ABS-KEY ( crystal ) OR TITLE-ABS-KEY ( xp ) OR
TITLE-ABS-KEY ( “extreme programming”) )
```

2. Restriction by field of study: if the data source offered an option to filter results to only computer science related studies, the initial list of results was filtered (see Figure 7).

The search strings were run in the selected sources. The execution of this searches provided a total of 1451 results. Table 3 shows the results obtained after executing the adapted search queries.

Table 3 Searh results overview

Search space	Search results	Filtered results
ACM	110	110
Google Scholar	937	937
IEEE Xplore	61	61
ISI Web of Science	1259	223
Science Direct	24	10
Scopus	554	110
All Libraries	**2945**	**1451**

An initial set of relevant studies was selected by reading the title and abstract of all the studies obtained from the web search engines, and these were evaluated according to the inclusion and exclusion criteria. The studies were eligible for inclusion if they fulfilled the following criteria: all kinds of studies related to the research topic will be selected if, after analyzing title and abstract of the document, it is evident that the study presents some kind of creative approach or technique for gathering requirements in agile software development. All documents that did not fulfill these criteria were discarded.

It was found that, of the 1451 studies collected, only 51 were relevant, i.e., 4 per cent. The number of relevant studies that were found in each digital library is presented in the third column of Table 4. These numbers are detailed as percentages in the 4th and 5th columns.

There is a noteworthy difference between the number of results obtained and the number of relevant results. It was observed that even though many of the studies that were acquired during the search contained some words from the search query, their scope was not related to the research questions presented in this systematic literature review (Section 3.2) and they were, therefore, excluded. In addition, it is worth mentioning that many of the collected works were related to creative approaches but not within the specific context of agile development or to requirements analysis in particular.

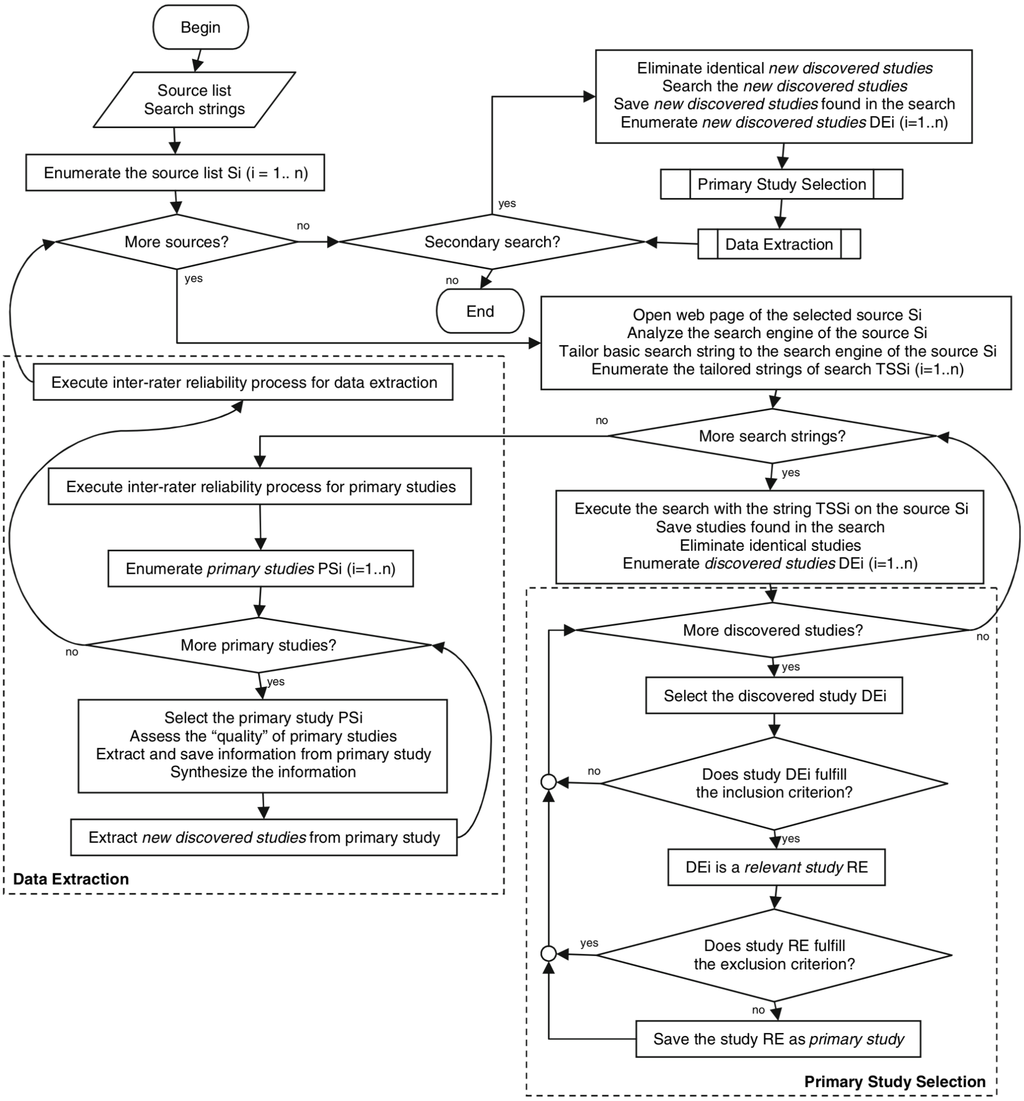

Begin
Source list
Search strings
Enumerate the source list Si (i = 1.. n)
More sources?
no
yes
Secondary search?
yes
no
End
Eliminate identical *new discovered studies*
Search the *new discovered studies*
Save *new discovered studies* found in the search
Enumerate *new discovered studies* DEi (i=1..n)
Primary Study Selection
Data Extraction
Open web page of the selected source Si
Analyze the search engine of the source Si
Tailor basic search string to the search engine of the source Si
Enumerate the tailored strings of search TSSi (i=1..n)
More search strings?
Execute the search with the string TSSi on the source Si
Save studies found in the search
Eliminate identical studies
Enumerate *discovered studies* DEi (i=1..n)
More discovered studies?
Select the discovered study DEi
Does study DEi fulfill the inclusion criterion?
DEi is a *relevant study* RE
Does study RE fulfill the exclusion criterion?
Save the study RE as *primary study*
Primary Study Selection
Execute inter-rater reliability process for data extraction
Execute inter-rater reliability process for primary studies
Enumerate *primary studies* PSi (i=1..n)
More primary studies?
Select the primary study PSi
Assess the “quality” of primary studies
Extract and save information from primary study
Synthesize the information
Extract *new discovered studies* from primary study
Data Extraction

Fig. 6 Procedure for execution os the systematic review (from [61])

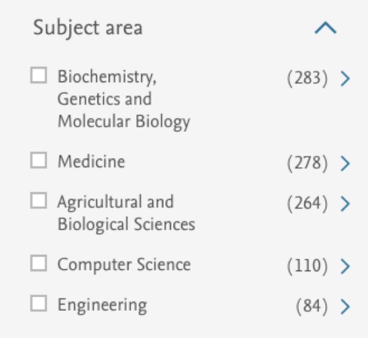
Table 4 Details of search results

Search space	**Search results**	**Relevant studies**	% of relevant studies	% of all relevant studies
ACM	110	7	6%	14%
Google Scholar	937	17	2%	33%
IEEE Xplore	61	6	10%	12%
ISI Web of Science	223	9	4%	18%
Science Direct	10	0	0%	0%
Scopus	110	12	11%	24%
All libraries	**1451**	**51**	**4%**	**100%**

It is important to consider that one specific study may be found in several search spaces at the same time. This is denominated as a relevant duplicated study and should be filtered. Once the duplicated studies had been removed, 17 relevant studies remained (see Table 5). None of these studies fulfilled the exclusion criteria and they were, therefore, all selected as primary studies. Appendix I contains the full list of the primary studies.

3.5 Studies quality assessment planning

In order to avoid biases and ensure the relevance of the selected studies, we undertook a quality assessment. Since stan-



Subject area

□ Biochemistry, Genetics and Molecular Biology (283)

□ Medicine (278)

□ Agricultural and Biological Sciences (264)

□ Computer Science (110)

□ Engineering (84)

Fig. 7 Example of Filter Option in Scopus

Table 5 Overview of relevant studies

Relevant studies	**# of studies**	**Percentage**
Relevant studies	51	100%
Duplicated relevant studies	34	67%
Non-duplicated relevant studies	**17**	**33%**

dard quality criteria that are valid for all systematic reviews does not exist, we compiled questions from the suggestions proposed by Kitchenham in [38]. The proposed checklist in this systematic review assesses the individual studies in terms of rigor, credibility and relevance (see Table 6, 7 and 8).

3.6 Data extraction and synthesis planning

According to the guidelines of Kitchenham [37] and Biolchini et al. [7], certain information needs to be extracted from each of the selected studies. To facilitate this task, we employed the Mendeley software to extract basic information, such as the title, authors, publication date, DOI, type of work and abstract.

We additionally collected a template for each primary study by adapting the proposal from Pino et al. [61] to collect the most important characteristics of each study (see Table 9).

Besides this basic information, we extracted relevant information that answered each of the research questions (see Table 10).

Furthermore, in order to facilitate the synthesis of the studies, we collected information regarding the problems identified by the studies, the objectives, the methodology employed, the conclusions and the future work (see Table 11).

4 Data extraction

This section summarizes the most important data that has been extracted from the primary studies. A classification of the studies and a quality assessment of them will be presented later in this paper.

4.1 Overview of studies

This section summarizes the results of the review of the primary studies.

4.1.1 Maiden and Hollis

In [S1] the authors propose to integrate creative thinking into agile processes in order to generate new and useful requirements.

The authors first question the effectiveness of agile processes, which very often miss some requirements by just thinking of the simplest solution. Based on previous studies [59], they claim that agility is not always sufficient to generate the requirements needed to innovative new software solutions. One of the reasons for this is that the short duration of a sprint may discourage creativity.

To respond to this problem, they present an extension of the Ambler agile process [2]. This extension consists of two sub processes that introduce creative techniques, as seen in Figure 8. The first creative sub process is included during the envisioning process in sprint zero, between the plan and the evaluation phases. The second creative sub process takes place at the beginning of some sprints, when the epics with the most creative potential are selected and become the focus of the creative activities. During these creative phases, short creativity workshops are run to discover more new ideas. The idea of using creativity workshops to support the creative process in requirements engineering had already been investigated by the same authors in previous studies [51,86, 50,52,35].

These creative workshops employ different creative techniques that have been, to a great extent, inspired by or extracted from the works of Higgins and Michalko [30,56].

This work follows a case study research methodology and provides evidence that the method presented could be used in real software projects. However, although several projects in which the technique was applied are mentioned, only one is documented, which consisted of the redesign of a television listing website at BBC Worldwide. The results of the experiment suggest that the use of creative techniques to elicit requirements can produce more new requirements. These requirements, however, tend to be seen as less useful, especially in the initial phases. For these reasons, the authors recommend using this type of technique in the seeking step of projects rather than to produce incremental change.

For future works the authors recommend the use of lightweight creativity techniques in other agile projects.

Table 6 Quality assessment - Rigor

Category	Item	Assessment criteria	Score	Description
Rigor: Has a thorough and appropriate approach been applied to key research methods in the study?	QA1	Does the paper include a clear statement of the aims of the study?	2	Yes, aims of the study are clearly described
			1	Partially, aims are described but unclearly
			0	No, aims are not described
	QA2	Does the study present a detailed description of the approach?	2	Yes, the approach is well described and can be applied
			1	Partially, the approach is difficult to understand and to replicate
			0	No, details are missing
	QA3	Does the study follow a systematic methodology that can be applied in another setting?	2	Yes, if follows a systematic methodology
			1	Partially, if follows a methodology but it is not fully described
			0	No, it does not follow a systematic methodology

Table 7 Quality assessment - Credibility

Category	Item	Assessment criteria	Score	Description
Credibility: Are the findings well-presented and meaningful?	QA4	Is the reporting clear and coherent?	2	Yes, the findings are clearly described
			1	Partially, the findings are described but are not easy to understand
			0	No, the findings are difficult to understand
	QA5	Is the research process been adequately documented?	2	Yes, the research process is fully documented
			1	Partially, some parts of the research process are omitted
			0	No, the research process is very inadequately documented
	QA6	Is the proposal validated?	2	Yes, it is validated
			1	Partially, validation is ongoing or data is not significant
			0	No, it is not validated
	QA7	Are the links between data, interpretation and conclusions clear?	2	Yes, results are clearly described
			1	Partially, results are described but not properly linked to data
			0	No, interpretations and conclusions are unclear or not described at all

Table 8 Quality assessment - Relevance

Category	Item	Assessment criteria	Score	Description
Relevance: How useful are the findings to the software industry and the research community?	QA8	Is knowledge or understanding been extended by the research?	2	Yes, the research provides new useful knowledge or understanding
			1	Partially, the study expands some knowledge from previous study
			0	No, the research does extend knowledge or understanding
	QA9	Is the proposal cited by other authors?	2	Yes, it is cited by 5 or more authors
			1	Very few, less than 5 articles cited the study
			0	No, no one cited the study
	QA10	Does the study present a proposal that can be replicated in other organizations/settings?	2	Yes, the proposal can be replicated under different circumstances
			1	Partially, the proposal can be replicated with some restrictions
			0	No, the proposal cannot be replicated

Table 9 Data extraction - Characteristics of primary studies

Title	Title of the study
Authors	Authors of the study
Published in...	Journal, magazine or conference
Year	Year of publication
Abstract	Summary of the study

Table 10 Data extraction - Research questions

R1	What Creative Technique?
R2	What Empirical Evidence?
R3	What Benefits and limitations?
R4	Where is, or could be used?
R5	What Agile Methodologies?

4.1.2 O'Driscoll

In [S2], the author presents a technique named the Agile Design Data Modelling (ADDAM) process.

Table 11 Data extraction - Further information

Identified Problems	What problems identified by the authors exist in requirements elicitation?
Objective	What are the motivations and aims of the study?
Methodology	What research Methodology is being followed by the study?
Conclusions	What are the conclusions of the study?
Future work	What is the future work suggested by the study?

Motivated by the difficulties of requirements analysis to meet the needs and expectations of end-users, this solution aims to improve the requirements analysis process and the value of the solutions developed by involving agile and design thinking in business oriented data modeling.

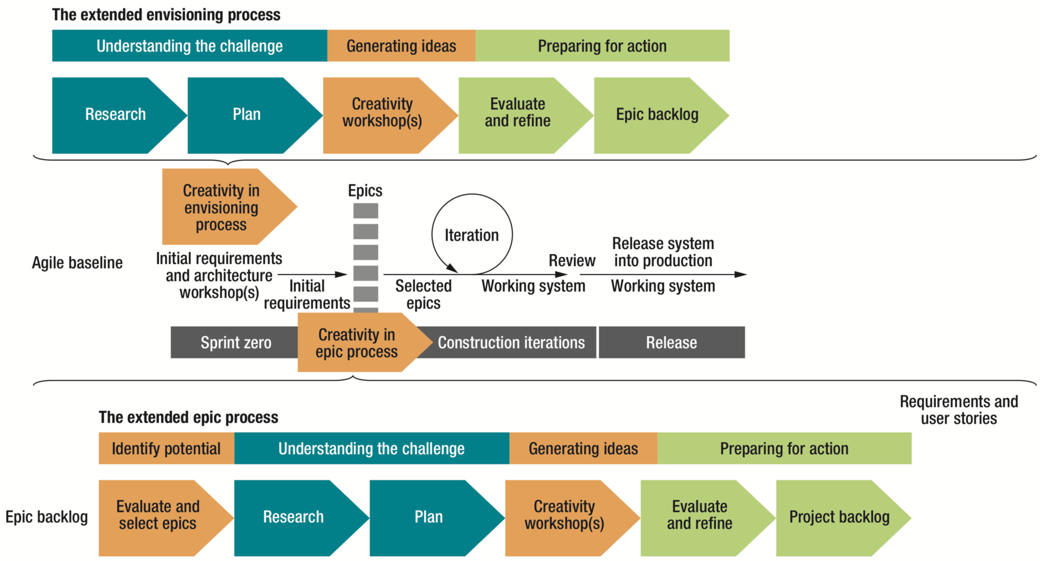


Fig. 8 The Extended Envisioning Process [S1]

The process, as shown in Figure 9, consists of five stages: Problem Formulation, Model Existing Application, Envisioning, Model Required Solution, and Evaluation.

This work documents an action research study which took place within an organization's IT Project between May 2014 and March 2015 (see Figure 10). No other empirical evidence or studies which apply this technique have been found.

The study suggests that this technique for requirements analysis can be effective at improving business stakeholders' engagement and building a better understanding of the business and its problems. This leads to cost savings and the development of solutions that better meet business needs. It also suggests that the data model presented improves communication between the business and IT groups and also enables faster and more efficient systems development. The study also reports that when it comes to Information Systems development, the non-technical factors around people and organizational culture are much more significant than the technological and data-related factors.

4.1.3 Gamble

The main idea behind the technique proposed in [S3] is to align and standardize metamodels in order to improve the link between design intent, business and development.

The author underlines that in many software development projects, despite the recent progress as regards adopting agile methodologies, the output does not always meet the business needs.

One of the reasons for this is that each stakeholder uses his or her own set of methods and tools, and these different frameworks are poorly aligned.

In order to respond to this problem, the author examines the different metamodels used by agile development, service design thinking and architecture frameworks and provides a proposal with which to align these metamodels.

He illustrates the technique by using the examples of the following frameworks: TOGAF (The Open Group Architecture Framework) for enterprise architecture, SAFe (the Scaled Agile Framework) [41] for agile enterprise development, and the concepts of service design thinking [81,46]. The result of linking these frameworks is the model presented in Figure 11. This approach offers a design linkage while simultaneously constraining the complexity to a level that is usable by developers, and traceable by architects.

Apart from the examples presented to explain the model, no further evidence is provided by the study. Although there is the potential to tie the development task to the design artifacts from architecting and design thinking activities, there is still very little guidance.

The suggested further work is to provide empirical evidence by testing the proposal in different situations of enterprise development.

4.1.4 Newman and Ferrario

The technique proposed in [S4] is based on the integration of design thinking into an agile and participatory framework. It does so by introducing user collaboration from the very beginning, even before requirements are elicited.

The authors emphasize that complex and undefined social problems cannot be addressed through the use of formal methods and analysis alone. Even though some works have already made use of creativity techniques like creativity workshops to elicit requirements [50,69], it has not yet

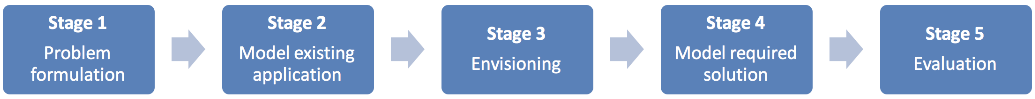


Fig. 9 Agile Design Data Modelling process

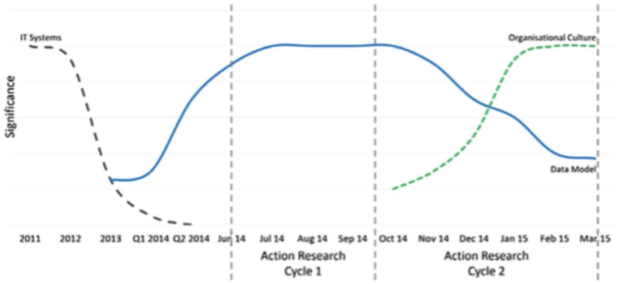


Fig. 10 Action research project overview [S2]

been investigated whether such approaches can be applied to groups with little or no experience in the problem domain. Similarly, the approaches that present artifacts as focal points for participants to use in the design process are not developed on the basis of feedback obtained from the stakeholders.

Motivated by these problems, the authors' purpose is to integrate design thinking into an agile and participatory framework that can be applied to complex and not well understood socio-technical problems.

This creative approach is called "Divingboard" and makes use of creative workshops and physical artifacts to allow participants to better engage with the problem context. An overview of the process can be observed in Figure 12.

This work provides empirical evidence of the technique presented in the form of a nine-month case study. This was carried out with a remote Scottish Island community in order to develop a renewable energy forecasting system. The results of the study show that this technique facilitates skill sharing, team building, creative problem exploration, the rapid prototyping of ideas and possible solutions for complex underspecified and open-ended problems. Nevertheless, it also underlines that users should be open minded and willing to discard and change ideas if the prototype proves unsuccessful.

The study concludes that creativity-driven workshops and physical artifacts are effective tools as regards encouraging the generation of requirements and solutions for complex problems. Furthermore, in this creative problem-solving technique, user participation is key aspect to ensure the success of the agile software development process.

4.1.5 Patton

The proposal presented in [S5] introduces a technique that consists of the usage of interaction design concepts in an agile development process to help better define requirements.

The author identifies that the use of agile methodologies, such as XP, tends to deliver high quality software quickly, but the resulting product does not impress customers or meet their expectations. Very often, the final product ends up offering features that the end user does not need while lacking features that are required.

In order to solve these issues, the author emphasizes that a better way in which to elicit requirements in agile development projects needs to be explored so as to develop cost-effective high-quality software that better meets users' expectations. The proposal presented consists of the introduction of collaborative Agile Usage-Centered Design sessions during the development process, in which developers, interaction designs, domain experts, business leaders, and the actual End-User, participate in "designing" the requirements.

This study employs an experience report to provide evidence of the incorporation of Agile Usage-Centered Design into the day to day work of agile projects so as to deliver high quality software. It contributes with a detailed description of the experimental approach and offers a series of guidelines.

Based on the experience at Tomax Technologies, this approach appears to improve the stakeholders' understanding and ownership of the software. The paper also reports that priorities are easier to identify by looking into user roles and their focal task cases. In addition, Agile U-CD as an instance of Interaction Design is simple, teachable and repeatable. All these advantages make it possible to deliver solutions on time with high quality and better meet user expectations. As a counterpoint to these advantages it is necessary to consider that this approach requires constant collaboration, which can be exhausting.

The author comes to the conclusion that while agile development methods allow high quality software to be delivered sooner, adding interaction design concepts helps to increase confidence and user empathy that better leads to end-user satisfaction. Although the experience presented occurred within an extreme programming methodology, this approach also feeds other agile development methods well.

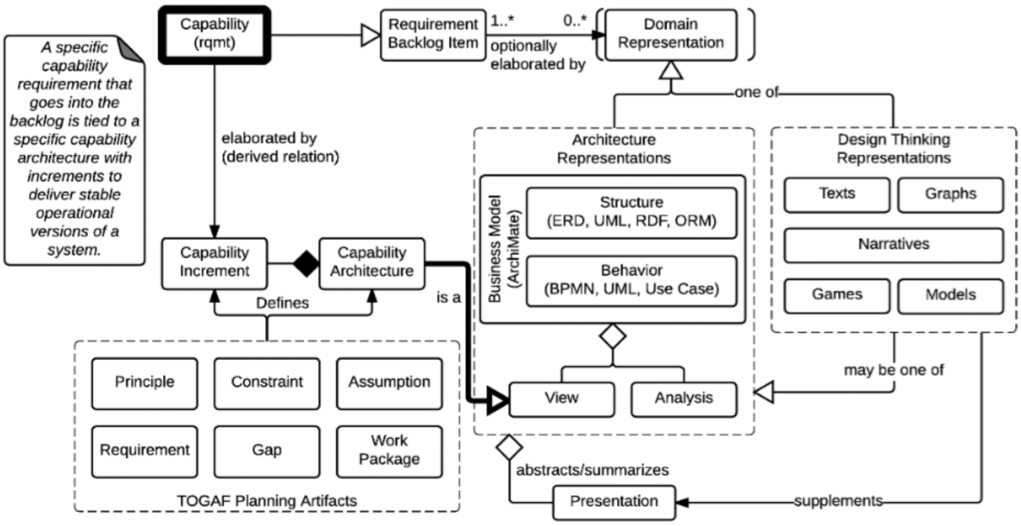

A specific capability requirement that goes into the backlog is tied to a specific capability architecture with increments to deliver stable operational versions of a system.
Capability (rqmt)
Requirement Backlog Item
1..*
0..*
optionally elaborated by
Domain Representation
one of
elaborated by (derived relation)
Architecture Representations
Design Thinking Representations
Business Model (ArchiMate)
Structure (ERD, UML, RDF, ORM)
Behavior (BPMN, UML, Use Case)
Texts
Graphs
Narratives
Games
Models
Capability Increment
Capability Architecture
Defines
is a
Principle
Constraint
Assumption
Requirement
Gap
Work Package
TOGAF Planning Artifacts
View
Analysis
may be one of
abstracts/summarizes
Presentation
supplements

Fig. 11 ADDAM Process detailed description [S3]

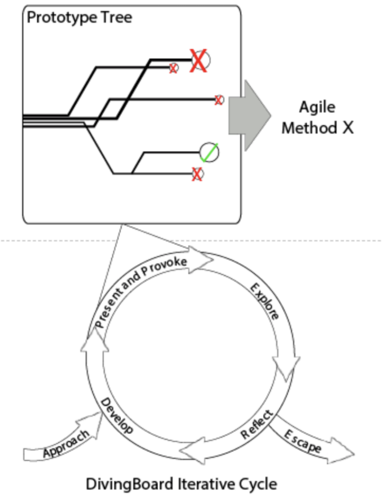


Fig. 12 Divingboard process [S4]

4.1.6 Percival, Braz and Adilson

In [S6], the IBM Design Thinking method is presented. This method extends the original design thinking principles to agile software development by providing a new approach with which to capture requirements that includes end-user feedback during all the project development phases.

The authors claim that traditional design thinking is insufficient to be integrated into an agile software development process. Software development demands a close relationship between designers and engineers, and traditional design thinking separates design from implementation.

Considering this, the study presented establishes two main objectives. Firstly, that of providing an easy description of the IBM Design Thinking method and its differences from traditional design thinking. Secondly, that of evaluating its benefits and limitations by means of a survey with the developers and designers of five real software development projects.

This IBM Design Thinking Method introduces three new elements into the traditional design thinking process: hills, sponsor users and playbacks. Hills are a new way in which to express user's needs, sponsor users are real human beings who can share their experiences and points of view and playbacks are checkpoints when the project team and Sponsor Users meet in order to review the state of the project and plan the next steps. The comparison of this method with traditional design thinking is presented in Figure 13 and the complete extended method is shown in Figure 14.

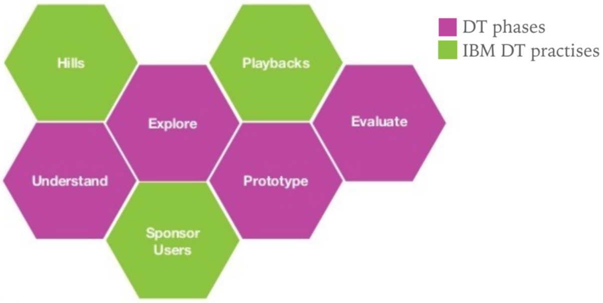


Fig. 13 IBM DT compared to traditional DT [S6]

The research methodology employed by this study is a survey that follows Barbara Kitchenham's guidelines for "Personal Opinion Surveys" [38].

The results of the survey evidence that this method can help the development team to attain a better understanding of the problems to be solved and what the best solutions by

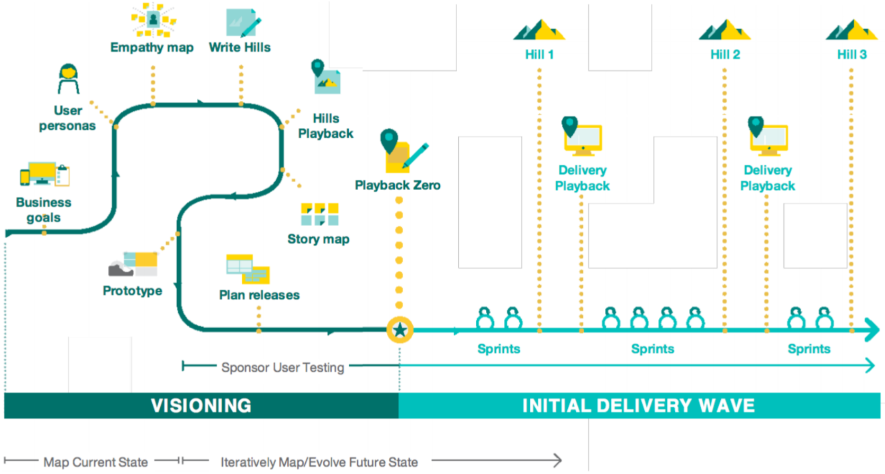


Fig. 14 IBM DT Software Development Framework [S6]

which to satisfy the user needs are. This deep understanding becomes solutions that are valuable for the End-users. However, it is necessary to consider that in order to apply this process, companies need to be willing to reorganize their teams, review their work model and functional roles and change their approach to solving problems.

The authors mention that this method could be used in incremental software development, such as cloud based software, and emphasize that further studies are needed to measure satisfaction levels accurately and to comprehend the limitations of the software framework developed.

4.1.7 Lombriser, Lucassen and Brinkkemper

In [S7], the authors present the Gamified Requirements Engineering Model, which integrates gamification and engagement theories in the context of requirement engineering performance.

The authors emphasize that despite the important role that requirements engineering plays in software development, insufficient stakeholder participation in requirement elicitation and poor collaboration and communication are still very common problems. This leads to low-quality and unsatisfactory requirements and consequently to the inability to deliver on time, within cost or expected scope.

The study aims first to improve the quality and increase the creativity in requirements, and second to evaluate the effectiveness of gamification so as to improve stakeholder engagement and productivity in requirements engineering when online digital platforms are used.

The GREM model consists of a relationship between three variables: gamification (defined as the application of game design elements in non-gaming contexts) [20], engagement (defined as the emotional, cognitive and behavioral connection that exists, at any point in time and possibly over time, between a user and a resource) [19,55] and performance.

Besides these three variables, two control variables are included in order to mitigate threats to internal validity: motivation and stakeholder expertise. Three sub-dimensions are defined for stakeholder engagement: emotions, cognition and behavior. Performance is sub-divided into productivity, quality and creativity, which are perceived as supportive concepts with which to measure the output in requirements elicitation. The relationships among all these concepts are shown in the conceptual model of Figure 15.

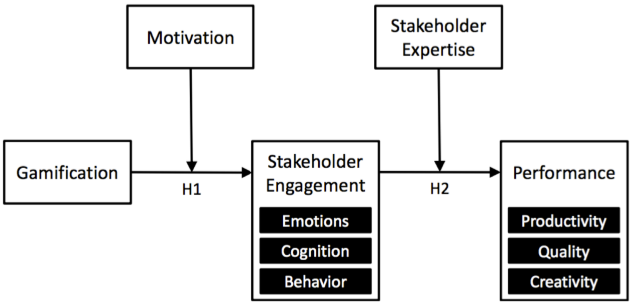


Fig. 15 The Gamified Requirements Engineering Model [S7]

This work follows a controlled experiment research methodology. It documents an experiment that took place in a business environment (IT consultancy company in Munich, Germany) with a total of 12 stakeholders divided in two groups (control and experimental group) well equilibrated in terms of gender motivation and experience.

The results of the experiment evinced that creativity improvement and performance increase in terms of the quantity and quality of the requirements. However, it was also observed that the success of gamification depends very much on the game elements and game mechanics. While rivalry elements seem to bring more creativity into requirements, they may not be suitable in phases in which cooperation and the exchange of ideas are needed. This leads the authors to a new hypothesis: "While requirements elicitation is positively supported by competitive game elements, cooperative game elements are more suitable for requirements analysis, specification and validation".

This leads to conclude that gamification can positively influence the elicitation process in agile requirements engineering. As future work the authors emphasize that more experiments with different sample sizes and game mechanics are needed to generalize the results and the applicability of the GREM.

4.1.8 Mahmud and Veneziano

In [S8], the authors present an approach with which to elicit and represent requirements within the SCRUM model based on the usage of mind-maps to establish the initial product backlog.

Mind Mapping is a technique that encourages people to think of, organize and represent information within a radial hierarchy, by locating the most important concept at the center of a given diagram and relate it to other concepts situated farther away from the center of the diagram [10].

The motivation to pursue this study originated from the difficulties involved in combining traditional requirements elicitation with agile approaches, and from the problems derived from agile requirement elicitation, such as poor customer involvement or costly system changes owing to the anticipation of too many upfront requirements.

The authors propose to confront these issues by evaluating how the requirements elicitation process within agile software development projects, and in particular within the SCRUM model, may benefit from the use of mind maps to develop a suitable product backlog. This new technique could also reconcile traditional approaches in software development with agile methods.

The authors carry out an experiment to evaluate whether the quality of requirements represented as a backlog product in SCRUM could be affected by the adoption of mind-mapping techniques during their elicitation and analysis process.

The results of the experiment show that the mind-mapping technique seems to support the analysis and capture of consistent and complete requirements very positively. However, it should be kept in mind that instructions and training on how to build mind-maps need to be provided to the stakeholders beforehand.

Of the validity limits of the experiment, the results show that by using a mind-mapping technique the overall quality of the product backlog is significantly higher if the product owners have little experience and not worse if the product owners are more experienced. This leads the authors to conclude that the mind-mapping and any derived technique could be used to set up the initial product backlog when developing with agile methodologies like SCRUM.

For future work, the authors suggest that more experiments and a larger amount of data are needed to obtain statistically significant results.

4.1.9 Maiden

In [S9], the author discusses the need to provide requirements analysts with a new creative and agile toolbox and the right instructions to use them. Although the focus is not on presenting a new creative technique or process for requirements engineering in agile development, these ideas are of great interest within the context of this systematic review.

The author emphasizes that traditional elicitation tools, such as observations, interviews and questionnaires, have some weaknesses. These methods, which rely on communication, tend to generate a lot of information but still omit some information that the analyst needs because, for example, certain knowledge is taken for granted.

The article does not present empirical evidence but is based on the author's subjective experience and argumentative research. He encourages his readers to think about new and more effective ways in which to elicit, acquire, and discover customers' requirements. The article suggests that new creative techniques shared among stakeholders, analysts and developers may diminish the weakness of traditional elicitation tools.

The author does not specify any particular type of project or organization that could benefit from this new toolbox, but since he mentions requirements analysts, it can be deduced that this new toolbox could be used on a wide variety of software projects.

The author concludes that analysts and stakeholders should look to agile development techniques and user-centered design for techniques such as the analysis of Web analytics, wireframing, and user stories and exchange their experiences and techniques with each other, not in paper form, but using social media.

4.1.10 Wanderley, Silveire and Araujo

The proposal presented in [S10] aims to give support to the modeling of requirements by means of a creative and agile technique based on the automatic transformation of mind maps into feature models based on model-driven engineering (MDE).

The authors emphasize that the formalization of requirements through modeling tools requires a technical knowledge that not all domain experts already have. Furthermore, many system requirements are not well understood, which increases the difficulty involved in producing conceptual models efficiently. On top of that, communication problems between domain experts and software engineers are very common.

All these reasons motivate the authors to present an approach that could facilitate and improve the domain analysis and modeling process. The main contributions of this work are the definition of a modeling environment that supports the creation of feature models by domain experts, independently of their knowledge of requirements engineering. An overview of the proposed solution can be seen in Figures 16, 17 and 18. In this study, the authors present an example of the approach with the tool TaRGet, whose objective is to automatically generate a suite of tests based on the specifications of cases written in natural language, but they do not provide any empirical evidence.

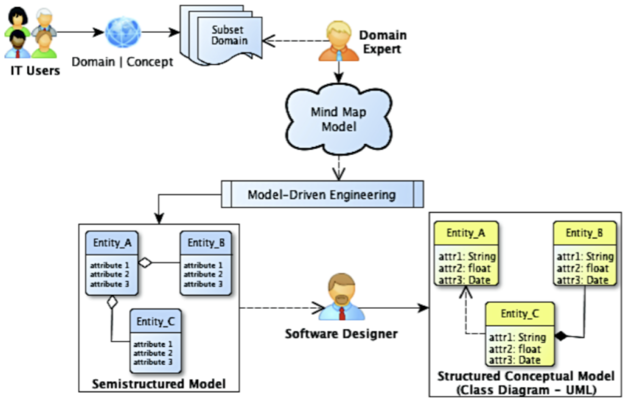


Fig. 16 Solution proposed by [S12]

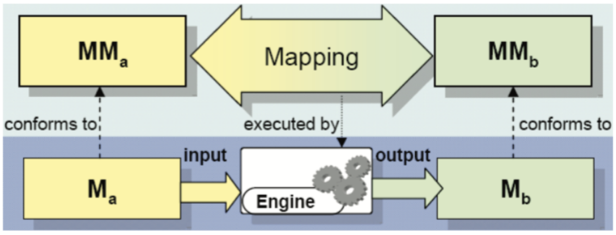


Fig. 17 Diagram of transformation among Models [S10]

The same authors continue this line of investigation in [S11], in which they present a study that aims to evaluate the use of creative requirements models together with goal-oriented models in a model-driven context. The approach establishes the mapping between the main concepts of the KAOS framework (which stands for "Keep All Objects Satisfied") and mind map models.

In a subsequent study [S12], the authors provide empirical evidence of the use of the initial model [S10] with two controlled experiments that involve senior, middle and junior software designers from industry and academia. The objective of these experiments is to evaluate the effort needed to build conceptual models expressed as UML class diagrams with the support of mind maps and to evaluate qualitatively the effectiveness of a mind map as regards understanding the domain analyzed.

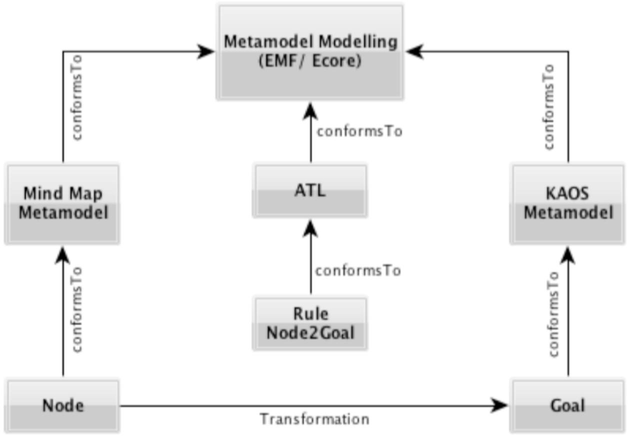


Fig. 18 Transformation process among models [S11]

The results of the experiments show that this approach simplifies the modeling process in an agile and creative manner by using mind maps, and facilitates the negotiation process related to the variability analysis of a domain. In other words, less effort is required to produce conceptual models through the use of mind maps than without them. However, in terms of productivity, the difference between the time required when using mind maps and when not using them is not meaningful.

The study also proves that it is possible to transform mind maps into feature models, without the need to refine the final model. This makes it possible to deduce that mind maps, with the extensions defined in their metamodels, can be used as an agile tool with which to model features.

As future work, the authors propose to extend and formalize the transformation rules with a specific ATL transformation language.

4.1.11 Bones, Harrison and Liu

In [S13], [S14] and [S15] the authors describe and investigate the use of a creative technique that can be used together with agile programming for requirements engineering.

The proposal is based on disciplined goal-responsibility modeling but introduces a non-formal and creative method with which to produce goal responsibility models under agile constraints: time, incompleteness and catching up after an initial creative burst.

Goal responsibility models represent the stakeholders' intentions for a system-to-be that will operate in an expected environment, in fulfillment of a contract. These models, when created in the very early phases of the project by combining all objectives and behaviors, are able to create realistic expectations in the minds of the stakeholders and are helpful when appraising the model for its feasibility, adequacy and testability. However, the creation of appropriate goal models can be complex when combined with agile constraints. This is particularly the case when stakeholders express their requirements as partial, hypothetical or functional designs, and in backlog driven projects which, after a few sprints, reach the point at which there are inadequate specifications for regression and acceptance testing.

These difficulties motivate the authors to investigate new creative techniques with which to produce goal responsibility models. In [S13], they describe goal sketching, a technique based on natural language that consists of the creation of a goal graph which expresses the high-level motivations behind the intention to develop the software. This initial graph is refined throughout a number of stages in a manner similar to the use of Scrum sprints [64]. During the refinement process, the goals are refined only as necessary for the current stage, which is called lazy refinement. The graph that is present during the process is called a "stage graph".

In a subsequent study [S14], the authors explore the use of UML diagrams such as activity diagrams (which are good at representing functionality and processes) to accelerate the speed of sketch drafting and refactoring.

They validate the proposal by employing an action research methodology and test the technique in both small and large development projects, such as the development of a product supported by venture capital, the specification, procurement and acceptance of a management information system for a food processing company and a tool to support professional services in healthcare.

Although the data obtained from the cases presented is not very significant, the results show that the technique can adapt to evolving requirements and cope with unresolved requirements. This leads the authors to conclude that this technique enables a successful development of evolvable systems and is in compliant with the real world needs of industrial software development.

Future work includes the application of the method to more industrial projects, the development of tools to accelerate sketching and refactoring, and export of goal sketches into KAOS, or the generalization of the method to enable the incorporation of other structured and UML modeling methods.

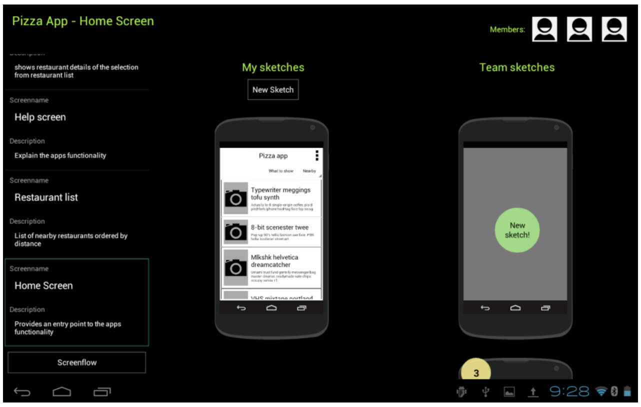
4.1.12 Hastreiter, Krause, Schneidermeier and Wolff

In [S16], the authors propose a collaborative tool for mobile prototyping that enables the user to sketch mobile applications with a set of basic, customizable objects and to share their prototypes with the rest of the team.

After performing an evaluation of existing sketching and prototyping tools for mobile application platforms by means of a competitive analysis, the authors highlight that, despite the importance of usability in aspects such as the efficiency of the system, effectiveness in the work process or the user's satisfaction, current software is not sufficiently attractive for a broader audience. Furthermore, existing mobile solutions do not support collaboration, which has been emphasized as an essential part of the development process. The analysis also states that the creation of complex interactive high-fidelity prototypes is not practical in a mobile work setting.

These problems motivate the authors to develop a requirements elicitation approach focused on UX methods to support action mode usage in order to leverage creative potentials.

The "Prime" proposal consists of a prototyping tool for android 10″ tablets that offers a low fidelity prototyping functionality (see Figure 19). The interface is composed of a drawing area into which objects can be dragged and dropped from a palette and be directly manipulated to fit the users' needs. Sketches and drawings can be incorporated into the prototype by means of the hardware camera. In order to enable collaboration between users, the tool includes an online backend based on parse.com (Parse Platform). This cloud platform provides easy to set up user accounts with different privileges and a database in which to store arbitrary objects.


Pizza App - Home Screen

shows restaurant details of the selection from restaurant list

Screenname
Help screen
Description
Explain the apps functionality

Screenname
Restaurant list
Description
List of nearby restaurants ordered by distance

Screenname
Home Screen
Description
Provides an entry point to the apps functionality

Fig. 19 Overview of the Prime platform [S17]

During the development process, the authors apply different UX techniques among small and multidisciplinary development teams practicing agile methods. One of them is SHIRA, "Structured Hierarchical Interview for Requirement Analysis", an interviewing technique that seeks to explore the meaning of abstract product qualities, such as "controllable", "simple", "impressive" or "innovative", for a specific software product in a specific context of use.

The results of the study show that this tool can speed up the development process and provide an adequate solution for the prototyping of mobile applications. It also improves user collaboration by allowing the users to share their prototypes with the rest of the team and maintains all team members updated.

As future work, the authors suggest the further optimization of the tool and the design of a method set with which to emphasize requirements and to evaluate ideas during the engineering process.

4.1.13 Sulmon, Derboven, Montero and Zaman

In [S17], the authors propose the User-Driven Creativity Framework, a participatory design methodology based on creativity theories that aims to stimulate end user creativity in the early design process of innovative applications with which to elicit user requirements.

The authors highlight that agile methods tend to assume that users can easily say what they want if asked and come up with innovative ideas. However, the reality is that gathering requirements takes lots of time and effort. Although it is claimed that agile methodologies promote collaboration between the customer and the development team, very few define how the people on the team that play the role of customer can learn what the real end user needs and how they can accurately represent those needs. This is the case of, for example, Scrum projects, in which the product owner is the person who represents the end user.

These problems motivate the authors to develop a framework as a viable alternative to enable requirements engineers or customer representatives to efficiently attain active stakeholder participation for initial requirements gathering within a limited amount of time. It consists of a 4-creative stage process, represented in Figure 20:

1. **Preparation Stage**: this consists of two activities. Firstly, an initial brainstorming activity with subject matter experts and other stakeholders serves to gather as much background information as possible. A session with actual end users then takes place. During this session, users are encouraged to actively engage with each other in small group discussions and analyze the relevant information. The session is divided into two parts, the first of which focuses on current practices in the domain, and the second of which focuses on the desired future practices.
2. **Incubation Stage**: this consists of leaving a break of at least three days between the first and second user sessions. This relaxation time offers participants "room for thought" and time to process the outcome of the first session.
3. **Inspiration Stage**: this is a co-design session, in which participants collaborate to create paper prototypes of the ideas gathered during the first user session. The results from this session can be used to understand users' needs and values and help designers and professional to finalize the design process.
4. **Transformation Stage**: this is the phase in which all the prototypes and information gathered are formally integrated by requirements engineers.

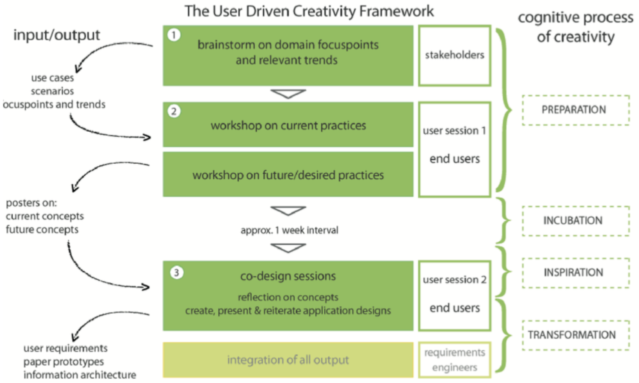


Fig. 20 The User Driven Creativity Framework

The results of the case study were rich and elaborate, and obtained in a limited amount of time. This proves that this framework can be successfully used to unveil high-level requirements and effectively yield the in-depth user knowledge and involvement required to establish a strong foundation for further agile development activities.

For future work, the authors underline the need for a refinement and an in-depth evaluation of the method. They also suggest that this framework could be applied in the design processes of other new application domains, in which a creativity-focused, user-oriented approach is appropriate.

4.2 Classification of studies

During the revision and extraction stage, in order to better quantify the information on each study, we defined a table with a classification synthesis for each research question (Tables 12, 13, 14, 15 and 16), with the corresponding values obtained by each of the primary studies.

4.3 Quality assessment

The results obtained after applying the quality criteria to each of the selected studies are shown in Table 18.

In terms of rigor, we found that 15 out of 17 (88%) studies obtained the maximum grade. Since most of the studies present very new approaches, it is in their self-interest to provide a clear description of the aims of the study (QA1),

Table 12 Synthesis of RQ1

RQ1: What methodologies or techniques that involve creative approaches in requirements elicitation (RE) in agile software development (ASD) exist?

	VALUES		**SYNTHESIS**																
	1	**0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
Process	YES	NO	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Framework	YES	NO	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Tools	YES	NO	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
Technique	YES	NO	0	0	1	0	1	0	0	1	0	1	1	1	1	1	1	0	0
Model	YES	NO	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Table 13 Synthesis of RQ2

RQ2: What empirical evidence of the application of creative techniques for RE in ASD exists?

	VALUES		**SYNTHESIS**																
	1	**0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
Experiment	YES	NO	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Action Research	YES	NO	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
Case of study	YES	NO	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1
Experience Report	YES	NO	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
Survey	YES	NO	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Table 14 Synthesis of RQ3

RQ3: What is it known about the benefits and limitations of using creative approaches for RE in ASD?

	VALUES		**SYNTHESIS**																
BENEFITS	**1**	**0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
Improvement of stakeholder engagement	YES	NO	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	1
Improvements in communication and ideas interchange	YES	NO	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
Improvement in novelty, quality and usefulness of rqmts	YES	NO	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
Better meet business needs	YES	NO	0	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	0
Ease problem understanding and rqmts elicitation	YES	NO	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
LIMITATIONS	**1**	**0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
Requirements are seen as less useful	YES	NO	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Users must be open-minded and willing to collaborate	YES	NO	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1
Reorganization of teams might be needed	YES	NO	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Training in novel techniques or process is needed	YES	NO	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0	0

their approach (QA2) and follow a systematic methodology (QA3), thus enabling other researchers or practitioners to test their proposals.

In terms of credibility, 8 studies (47%) qualified with the maximum grade. This represents less than half of the selected studies. We found that when documenting the findings (QA4) and the research process (QA5), some of the works did not provide any type of graphic representation that could facilitate an understanding of the results, or that some parts of the research process were omitted. With reference to validation (QA6), owing to the innovative nature of the approaches and the field of study, it was to be expected that not many studies would obtain the highest grade. A high percentage of the studies were still testing their approaches or the data provided was not significant.

To conclude our quality assessment, we evaluated the studies in terms of relevance. Here, 9 studies (53%) obtained the highest grade. With regard to the extension of knowledge (QA8), all the studies obtained the maximum grade since they proposed a new approach or extended understanding. With reference to the number of citations (QA9) it is not surprising that, owing to the innovative nature of the proposals they were not, on average, cited by many authors. In order to evaluate QA9, we effectuated a research in Google Scholar and counted the number of citations on the 20th of June of 2017. Finally, we evaluated whether the proposals can be replicated in other organization or settings, and most of the studies obtained a high grade in this respect.

4.4 Publications

This section provides a brief analysis of the studies in terms of year, country and type of publication.

4.4.1 Publication distribution per year

Figure 21 shows the distribution of the selected studies per year. Since the number of selected primary studies is not

Table 15 Synthesis of RQ4

RQ4. In what sort of organizations or projects using ASD could the use of these creative approaches be most suitable?

		VALUES			SYNTHESIS																
		1	**0**	**-1**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
Organization size	**Large Organization**	YES	NS	NO	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	0	0
	Startup or Small organization	YES	NS	NO	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
Duration	**Short development projects**	YES	NS	NO	0	0	0	0	0	-1	0	0	0	0	0	0	1	1	1	1	0
	Long-term development projects	YES	NS	NO	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Type of development	**Mobile Applications**	YES	NS	NO	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1
	Website development	YES	NS	NO	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Enterprise architecture	YES	NS	NO	0	0	1	0	0	1	0	0	0	1	0	0	1	1	1	0	0
	Web application	YES	NS	NO	0	0	0	1	0	1	0	0	0	0	1	1	1	1	1	0	0

Table 16 Synthesis of RQ5

RQ5. Which agile methodologies are being used to integrate creative techniques into agile software development RE processes?

	VALUES		SYNTHESIS																
	1	**0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
SCRUM	YES	NO	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Extreme Programming	YES	NO	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Agile Model Driven Development*	YES	NO	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1

**MDE itself is not an agile methodology, here is being considered as an application to agile for rapid conversion of the mind maps to formal models*

Table 17 Overview of quality assessment

Category	**Item**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**	**S15**	**S16**	**S17**
Rigour	QA1	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2
	QA2	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2
	QA3	2	1	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2
		6	5	6	6	6	6	6	6	0	6	6	6	6	6	6	6	6
Credibility	QA4	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	QA5	2	1	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2
	QA6	2	2	0	2	1	2	2	1	0	1	1	2	1	2	2	2	2
	QA7	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		7	5	6	8	7	8	8	7	4	7	7	8	7	8	8	8	8
Relevance	QA8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	QA9	2	0	0	2	2	0	2	2	0	2	2	1	2	2	1	1	1
	QA10	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2
		6	3	4	6	6	4	6	6	3	6	6	5	6	6	5	5	5
	Total	**19**	**13**	**16**	**20**	**19**	**18**	**20**	**19**	**7**	**19**	**19**	**19**	**19**	**20**	**19**	**19**	**19**

large, it is difficult to identify trends. We can, however, observe that all the selected studies are found within the period of the last 15 years, which gives evidence of the innovative nature of the research field.

4.4.2 Publication distribution per country and continent

Figures 22 and 23 show the distribution of studies per country. It is worth mentioning that if several countries were collaborating on the same study, we counted each country once.

It can be argued that the UK is by far the country with the largest amount of unique proposals and collaborations. India and the USA, which are also Anglophone countries that share a lot of synergies with the UK also have a great interest in this research field. It is notable that Portuguese speaking countries like Portugal and Brazil also provided a great number of papers on this topic. With regard to continents, we see that 70% of the studies were produced in Europe vs. 30% that were produced in the rest of the world, as shown in Figure 24. This indicates that Europe is currently a research driver within this field.

4.4.3 Publication type

In terms of type of publication, we have found that 71% of the studies were conference papers published in papers

Table 18 Overview of studies per quality assessment

Study	S4	S7	S14	S1	S5	S8	S10	S11	S12	S13	S15	S16	S17	S6	S3	S2	S9
Qualification	20	20	20	19	19	19	19	19	19	19	19	19	19	18	16	13	7

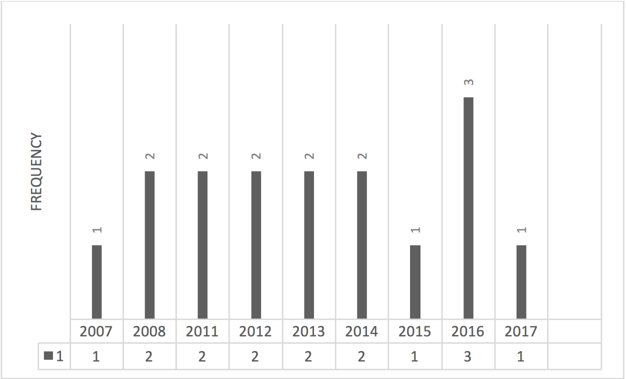


Fig. 21 Distribution per year of primary studies

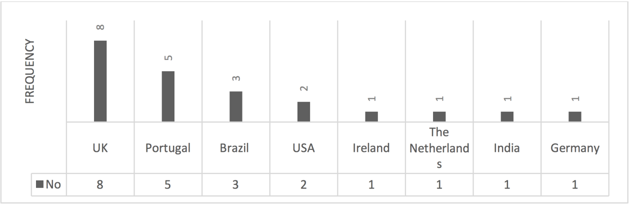


Fig. 22 Publication distribution per country of primary studies

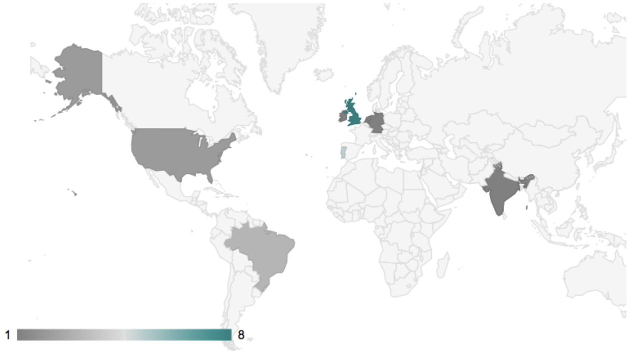


Fig. 23 Publication distribution per country and continent of primary studies

called "proceedings". These types of publications are generally used to present new ideas since it takes less time to get them accepted by conferences than to get them published in journals.

While some decades ago it was difficult to find conference proceedings, and researchers used to extend their conference papers into journal versions,, this trend now seems to have changed owing to the current frenetic development in the era of digitalization.

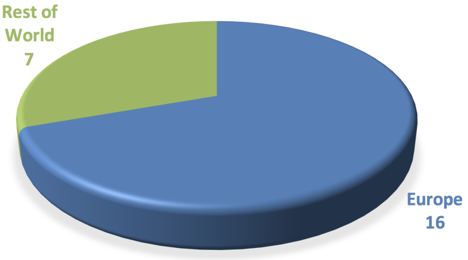


Fig. 24 Publication distribution of primary studies Europe vs Rest of World

As commented on in the paper "From Conference Papers to Journal Papers: Challenges and New Ideas" [39] "At present, for an extended version of a conference paper to be accepted in a journal, a minor extension is not sufficient because that would amount to publishing the same paper twice. To be accepted, the extended version must contain a large amount of new material. Because of this requirement, most authors who submit papers to conferences are unable to come up with follow-up journal publications".

The data we have obtained shows evidence of this behavior. Figure Figure 25 summarizes the publication type distribution of primary studies.

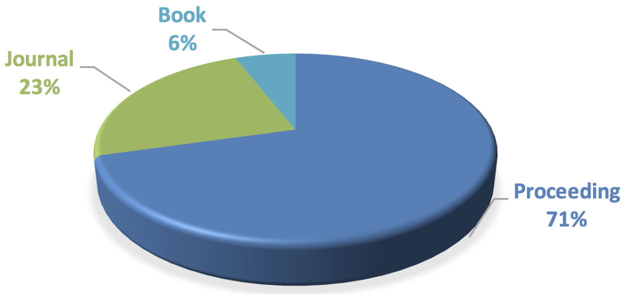


Fig. 25 Publication type distribution of primary studies

5 Data synthesis and results

This section describes the results of the present systematic review. We analyze the studies on the basis of the proposed quality assessment and research questions proposed.

5.1 Results of the SLR

This section discusses the answers found to respond to the research questions proposed during the planning phase of the systematic review, listed in Section 3.

This systematic review identified 17 primary studies. We found 13 unique approaches in the 17 identified. Studies [S10] [S11] [S12] and the studies [S13] [S14] [S15] belong to the same approaches, respectively. For this reason, in the review tables (19 to 26), we shall refer to S10, S11 and S12 as [S10] and S13, S14 and S15 as [S13].

RQ1. What methodologies or techniques that involve creative approaches in requirements elicitation (RE) in agile software development (ASD) exist

To answer RQ1, we have seen that the selected studies present 13 different and unique proposals. According to the authors, we have found different ways of classifying the studies, and these are: “process”, “framework”, “tools”, “model” and “technique”. This is explained in greater detail in Table 19, while the graphic distribution is shown in Figure 26.

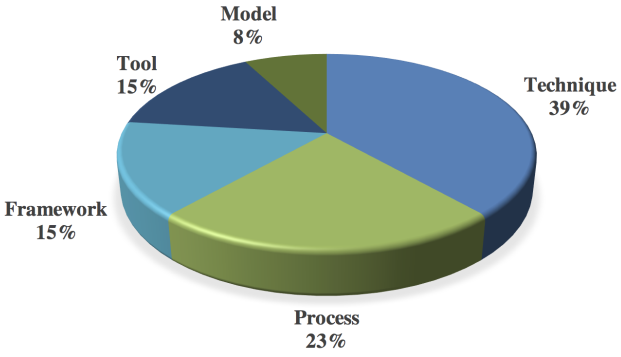


Fig. 26 Type of approaches proposed by the primary studies

We were unable to find two different research groups proposing or testing the same approach. We have, however, identified common design methodologies or frameworks that have been combined with other techniques or extended to be integrated into requirements elicitation and agile development as a basis on which to elaborate the new approaches. These are detailed in Table 20 and are presented in a graphic manner in Figure 27.

Additionally, we have identified common creative techniques that are used by these approaches to explore and elicit ideas. These are detailed in Table 21 and represented graphically in Figure 28.

RQ2. What empirical evidence of the application of creative techniques for RE in ASD exists

To respond to this research question, we firstly looked at the type of research provided by the selected studies. We describe this in detail in Table 22 and represent it graphically in Figure 29.

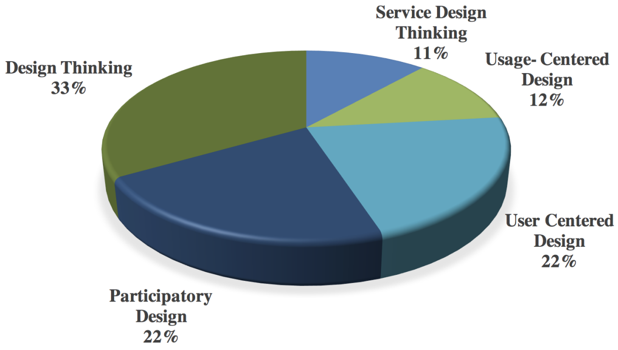


Fig. 27 Design methodologies

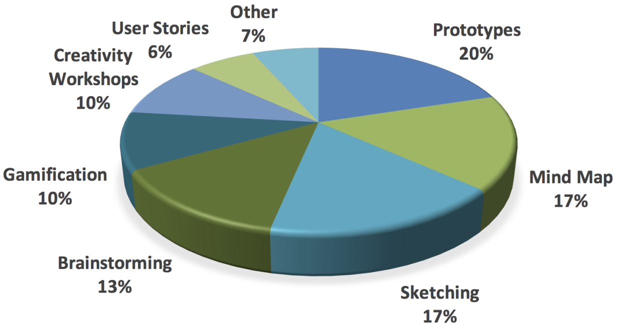


Fig. 28 Overview of creative techniques employed by the studies

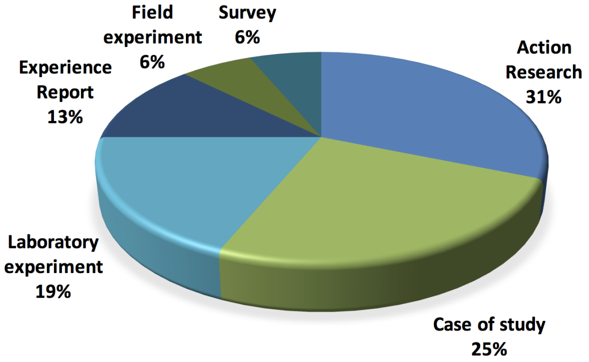


Fig. 29 Type of research employed by the primary studies

If we had looked at the empirical evidence of each of the studies individually using the criteria proposed by Kitchenham [37] and detailed in Table 23 as a basis, all the selected studies would have received a score of “0” and the evidence would have been considered “poor”. The reason for this is that most of the studies have only tested their approaches in one or two projects.

Owing to the innovative nature of the research field we are studying, in order to evaluate the empirical evidence and

Table 19 Detailed synthesis of approaches

Category	Percentage	Studies	Description
Technique	31%	[S3] [S5] [S10] [S13]	A technique describes a way of carrying a specific task or procedure. We have found four approaches proposing techniques to improve the requirement elicitation process by integrating creativity. [S3] presents a technique of linking metamodels to design intent; [S5] propose the usage of interaction design concepts in agile development; [S10] propose a technique of mind mapping modelling for building feature models. [S13] goal sketching for agile requirement elicitation.
Process	23%	[S1] [S2] [S4]	A process describes an iteration of activities that need to be carried out to complete a mayor task. In software engineering, it can describe the whole development process or a specific sub-stage of the development life cycle. We have found three works proposing processes that integrate creativity into requirements elicitation in agile environments: "Extended Envisioning Process" [S1] "Agile Design Data Modelling Process" [S2] and "Divingboard" [S4].
Framework	15%	[S6] [S17]	In computer systems, a framework is often a layered structure indicating what kind of programs can or should be built and how they would interrelate. We have found three studies presenting frameworks: [S6] presents the "IBM Design Thinking Development Framework" based in the ideas of Design Thinking and agile processes, while [S17] proposes the "User Design Creativity Framework", which is based in Participatory Design and the theories of creativity.
Tool	15%	[S9] [S16]	Software tools are programs, utilities, libraries, and other aids, such as editors, compilers, and debuggers, that can be used to develop programs [1]. We have found two studies that propose tools as a means of facilitating interchange and innovation in requirements elicitation. [S9] propose an agile toolbox to share RE techniques, while [S16] propose a mobile prototyping tool to elicit requirements.
Model	8%	[S7]	A model is a mathematical or graphical representation of a real-world situation or object [1]. We have found one study proposing a model, [S7] "Gamified Requirements Engineering Model", which integrates gamification in requirement elicitation process as a means of improving the novelty, quality and usefulness of requirements.

Table 20 Detailed synthesis design methodologies

Term	Percentage	Studies	Description
Design Thinking	23%	[S2] [S4] [S6]	A Methodology used as a non-lineal problem-solving approach to solve complex problems and focused on the users and their needs, which aims to ensure that the developed solution meets a real user need [45].
Service Design Thinking	8%	[S3]	A modern interpretation of the design thinking movement in the context of providing s services, most from businesses to customers [S3].
User-Centered Design	15%	[S9] [S16]	Is an approach to systems design and development that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques [21].
Usage-Centered Design	8%	[S5]	An approach touser interface designbased on a focus on user intentions and usage patterns. It analyzes users in terms of the roles they play in relation to systems and employs abstract use casesfortask analysis. The term was suggested by software developer and professor Larry Constantin and Lucy Lockwood [16].
Participatory Design	15%	[S4] [S17]	An approach to design in which the people destined to use the system play a critical role in designing it [71].

respond to our research question RQ2, we prefer to consider the evidence as a whole, rather than evaluating each of the studies individually (as proposed by Kitchenham [37]).

We have seen that these works provide both real world and laboratory evidence and are being tested in both short and long-term projects. Since between 10 and 20 projects have provided some sort of evidence of applying creative approaches to elicit requirements in ASD, we can conclude that the empirical evidence of applying creative approaches is moderate.

Moreover, many studies mentioned that their approaches are still being tested or would need further evidence under different circumstances.

RQ3. What is it known about the benefits and limitations of using creative approaches for RE in ASD?

Table 21 Detailed synthesis of creative techniques

Technique	Percentage	Studies	Description
Prototypes	46%	[S3] [S4] [S5] [S6] [S16] [S17]	Prototype is a preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system [14].
Mind Map	23%	[S3] [S8] [S10]	Mind map is a diagram used to visualize and organize information [S8].
Sketching	23%	[S3] [S4] [S13]	Sketching is a rapid drawing technique [S3].
Brainstorming	31%	[S1] [S4] [S5] [S6]	Brainstorming is a process where participants from different stakeholders groups engage in informal discussion to rapidly generate as many ideas as possible without focusing on any one in particular [87].
Games / Gamification	23%	[S3] [S6] [S7]	Gamification is the application of typical elements of game design elements (e.g. point scoring, competition with others, rules of play) in non-gaming contexts [20].
Creativity Workshops	23%	[S1] [S4] [S17]	Workshops that empower creative thinking [S1].
User Stories	15%	[S7] [S10]	Description of a feature written from the perspective of the person who needs this. It consists of a written text, conversation about it, and acceptance criteria [87].
Random Start	8%	[S1]	Adopted from Hall of Fame, M. Michalko [56], propose the use of characters to force connections to generate new requirements according to theme.
PICL Technique	8%	[S1]	Proposed by Higgins [30] consist of selecting 25 random short statements.
Storyboards	8%	[S6]	Sequence of pictures that represent the workflow of the user [87].

To respond to RQ3, we analyze the benefits and limitations that are particularly mentioned in the studies or that it is possible to infer from the information that has been provided. With regard to the benefits, we have identified five common benefits among the studies, which can be categorized in three different areas, as detailed in Table 24.

Regarding the limitations, it is worth mentioning that very few studies specifically discussed the limitations of their approaches. Nevertheless, we were able to identify the following limitations, commented on in detail in Table 25.

RQ4. In what sort of organizations or projects using ASD could the use of these creative approaches be most suitable?

It is worth mentioning that during the execution of the present systematic review we noticed that RQ4 might be too ambitious as regards the innovative nature of the area we are researching. Since the selected studies present only one or two pieces of evidence related to projects, it is difficult to determine their suitability in certain projects or organizations. The information provided in Table 266 is, therefore, based only on the information from the projects commented on in the studies and this does not necessarily mean that one approach is suitable only for these types of projects.

We have followed the following strategy: based on the information gathered from the review of the studies, we identified and defined three categories that could help us to respond to this research question:

1. *Organization size*: small (startup or projects with less than 10 participants) vs. large organizations (with 10 or more participants).
2. *Duration of the project*: short-term (duration of less than one year) vs. long-term (duration of more than one year).
3. *Type of development*: mobile applications, website development, enterprise architecture and web application.

After evaluating the studies, we obtained the following results, as presented in Table 26. It is important to note that the sum of percentages within one category (e.g. large + small organizations) is not necessarily 100%. The reason for this is that we evaluated the categories individually. One specific approach might have been successfully implemented in both large and small organizations.

RQ5. Which agile methodologies are being used to integrate creative techniques into agile software development RE processes?

Although very few studies mention a particular agile methodology being used to integrate these approaches, we found two works specifically mentioning the use of Scrum or employing traditional Scrum elements like sprints and product backlog [S1], [S8]; one, the use of Extreme programming [S5]; and four mentioning agile model driven development [S10], [S11], [S12], [S17]

We have found that most of the works prefer to propose approaches that would be applicable with different agile methodologies instead of restricting it to a specific one.

Table 22 Detailed synthesis of type of research

Type of research	Percentage	Studies	Description
Action Research Study	23%	[S2] [S13] [S16]	This type of research is characterized by producing relevant research results because it is grounded in practical action, aimed at solving an immediate business problem [5]. However, one of its disadvantages resides in the restriction to a single project and organization, this produces lack of control over variables, openness of interpretation, biases and omissions in the description and competition of objectives [12].
Case of study	31%	[S4] [S10] [S11] [S17]	Case studies are descriptive reports that provide in-depth analysis of a particular project or episode, usually within a single organization [12]. Similarly as Action Research, by restricting the research to a single organization, the generalization of results is difficult.
Laboratory experiment	23%	[S7] [S8] [S12]	This type of research controls and study a small number of variables intensively and aims to identify relationships between specific variables using quantitative and analytical techniques [12]. The problem of this approach lies in the limitations of the identified relationship to the experimental situation. Real world situations are much more complex and present different variables that the ones that can be studied with a controlled experiment.
Experience report	15%	[S3] [S5]	Experience reports are papers written by the people who participated in the work completed in a context of a real-life [12]. While [S3] describes an isolated experience, and provides merely anecdotal evidence, [S5] reports the results of using the proposed approach in real-world cases during at least some years.
Field experiment	8%	[S1]	Field experiments are an extension of the laboratory experiments into the real-life situations of organizations and/or society. They provide greater realism than other approaches but lack of sufficient control to enable replication [12]. Additionally to the documented experiment in [S1], the author mentions that after the first experiment the qualitative data from extending it to a second agile project did support the results and the proposal is currently being used in other development projects.
Survey	8%	[S6]	Surveys are snapshots at a particular point in time from which relationship inferences are made using quantitative analytical techniques [12]. The authors mention that the proposed methodology was applied in five real software development projects. Surveys are time-stamped samples from which relationship inferences are made using quantitative analytical techniques. They help to describe real-world situations and are appropriate to generalize, but cant ask about variables not yet recognized and may present bias in participants or researchers.

Table 23 Empirical Evidence Criteria evaluation by Kitchenham

Criteria	Number of projects	Quality	Score
The size of the within-company data set, measured according to the criteria presented below. Whenever a study used more than one within-company data set, the average score was used	Less than 10 projects	Poor	0
	Between 10 and 20 projects	Fair	0.33
	Between 21 and 40 projects	Good	0.66
	More than 40 projects	Excellent	1

5.2 Final Remarks

This systematic review provides an overview of the existing approaches that empower creativity in requirements elicitation within agile software development.

We have found that agile methodologies like Scrum, Extreme Programming or methodologies based in rapid modelling are preferred to introduce creativity in requirements elicitation.

Regarding the empirical evidence, due to the novelty of the research field, we have not found a strong evidence. However, considering the results that we have obtained with this systematic review, empowering creativity into requirement elicitation is not any more a theoretical idea, but a mixture of techniques, tools and processes that have been and are being successfully tested in the industry.

Due to the novelty and variety of approaches, which are mostly only tested in one or two projects, it is complicated to conclude in what sort of projects these techniques might be more suitable. However, it has been observed that highly interaction nature development projects like mobile or web

Table 24 Detailed synthesis of benefits

Benefit	Percentage	Studies	Description
Improvement of stakeholder engagement	31%	[S5] [S10]] [S16] [S17]	Refers to techniques or approaches that intensively encourage the participation of stakeholders.
Improvement in communication and ideas interchange	31%	[S2] [S4] [S16] [S17]	Refers to approaches that facilitate and improve communication between stakeholders, thus facilitating the interchange of ideas and productive collaboration.
Improvement in novelty, quality and usefulness of requirement	23%	[S1] [S7] [S17]	We have found only three studies specifically mentioning improvement in novelty and quality of requirements. Since evaluating innovation or quality is not an obvious and easy task, it might have happened that many of the selected studies did not specifically focused in measuring innovation or quality.
Better meet business needs	54%	[S2] [S5] [S6] [S9] [S10] [S13] [S16]	Almost all studies provided approaches that by empowering creativity into requirements elicitation the developed product better meet the demands of users and stakeholders.
Facilitate the understanding of the problem	69%	[S3] [S4] [S5] [S6] [S9] [S10] [S13] [S16] [S17]	It has been observed that all studies aim to deal with problem understanding related issues. By facilitating the understanding of the problem, requirement elicitation task will be more manageable and oriented to the exactly needed goals.

Table 25 Detailed synthesis of limitations

Limitations	Percentages	Studies	Description
Requirements obtained are seen as less useful at the very early stages	8%	[S1]	One study mentioned that during the initial stages, requirements obtained with creative techniques were seen as less useful. Nevertheless, in later stages after incubation of these ideas, their potential usefulness appeared to increase [S1].
Users should be open-minded, willing to collaborate and patient	31%	[S4] [S5] [S16] [S17]	In these approaches users are required to collaborate intensively, what may produce fatigue.
Reorganization of teams might be needed	8%	[S6]	One study specifically mentioned that in order to apply the proposed approach, teams and departments within the organization might need to be reorganized. This is a limitation because many organizations wont allow or facilitate this.
Training in novel techniques would be needed	69%	[S1] [S2] [S3] [S4] [S5] [S6] [S8] [S10] [S13]	Although not always specified, we have estimated that the studies presenting novel techniques that require some sort of training would need to consider this limitation in terms of time (needed to train the stakeholders) and other resources.

applications are good candidates to make use of these creative approaches.

Finally, we have found that, although creativity is an important ingredient to bring innovation, is not always sufficient to generate novel requirements and it needs to be accompanied by user engagement and a specific context where proper conditions like flexibility, time or resources have to be met.

6 Conclusions and further work

This Section summarizes the conclusions and future work of this article. To accomplish this, we analyze the achievement of objectives defined in Section 1 and the future work that has been identified. Additionally, we document and analyze a questionnaire that was developed to get a better understanding of the actual situation in the industry.

6.1 Achievement of objectives

At the beginning of this study, in Section 1 we defined several sub-goals that were needed to achieve the main goal: *to study the state-of the art in the approaches that strength creativity in requirements elicitation within agile software development*.

We now analyze how and to what extend these goals have been met:

Sub-Goal 1 - *Research on background studies*
In order to achieve this goal, we have carried out a research on systematic literature reviews, with a special focus

Table 26 Detailed synthesis of suitability

Category	Percentage	Studies	Description
Organization size	Large 54%	[S1] [S2] [S4] [S5] [S6] [S10] [S13]	54% of the approaches that mentioned the sort of organization in which their approaches were being tested or could be used referenced large organizations.
	Small 15%	[S10] [S13]	On the contrary, only 15% were recommended for small organizations. One of the reasons of this could be that most of the approaches require intensive final user involvement, and very often this implies some costs that small organizations cannot always afford.
	Small and Large 15%	[S10] [S13]	Six studies (two proposals) were suitable for both, small and large organizations.
Duration	Short: 15%	[S13] [S16]	Since most of the proposals were tested during a reduced amount of time or just specified the duration of the testing which mostly involved only the phase of requirements gathering, it was difficult to evaluate the proposals in terms of the duration of the project. From the information that we could extract or infer we found 2 approaches (15%) being used or suitable for short projects.
	Long: 31%	[S2] [S4] [S5] [S6]	On the contrary, we found 4 studies (31%) being used in long-term development projects. We found one study [S6] mentioning that the approach would not be suitable for short-term projects.
Type of product	31% Mobile applications	[S6] [S10] [S16] [S17]	We found 6 studies (35%) that tested their approaches in mobile applications.
	8% Website development	[S1]	We found one study (6%) testing their approach in website development
	31% Enterprise architecture	[S3] [S6] [S10] [S13]	We found four approaches (31%) been tested in in enterprise architecture software
	31% Web applications	[S4] [S6] [S10] [S13]	And four approaches (41%) been tested in web applications.

on systematic literature reviews for software engineering. We have also researched the topic of requirements elicitation and agile methodologies (providing details of Scrum, Extreme Programming and Kanban). Finally, we have investigated creativity, firstly from a theoretical point of view, and secondly as regards the application of creativity in design (with a special focus on Design Thinking) and requirements elicitation. All this research has been documented in Section 2.

Sub-Goal 2 - *Definition of a criterion that will serve to select and evaluate relevant studies*

This goal has been fulfilled in Section 3 during the planning stage of the systematic review. Here, we defined a quality assessment criterion in terms of rigor, credibility and relevance. We also defined a classification criterion that would help us to better organize the proposals and summarize the results.

Sub-Goal 3 - *Execution of a systematic literature review*

To fulfill this goal, we executed the search strategy that was defined during the planning stage and identified 1451 articles, of which 17 papers were included in the review as primary study papers. In Section 3, we provide a detailed description and present an overview of the selected studies.

Sub-Goal 4 - *Review and summary of the selected studies and identification of trends*

Finally, in order to accomplish this goal we carried out a detailed review of each of the selected studies, as documented in Section 4. Later, in Section 5 we analyzed and classified the information gathered.

6.2 Insights from the industry, a questionnaire

In order to attain a better understanding of the current situation in industry and collect the opinions of experts, we created a short web-based questionnaire.

The target group consisted of software development practitioners, and the respondents were selected by means of personal contacts and posting the questionnaire in software engineering social media groups. The sample can, therefore, be described as convenience sampling.

This questionnaire is available at:

`https://easyform.typeform.com/to/lu8StQ`

6.2.1 Questionnaire results

We collected responses from 75 individuals and defined the following question: Is there any relationship between the size of a team, the duration of a project or a requirements engineer's experience and the knowledge and use of creative approaches?

To evaluate this, we created several dynamic tables, crossing "team size", "project duration" and "years of experience" with "knowledge", "use" and "reception" of creative approaches, respectively.

Each of the results obtained is described below.

Team Size vs Knowledge of Creative Approaches

According to the results of the questionnaire, it would appear that small teams (1-10 team members) are more prone to know about creative techniques for requirements elicitation, followed by medium teams (10-50). Of these groups, the best-known creative approaches are "gamification", "mind-mapping" and "design thinking". Figure 30 shows the results obtained.

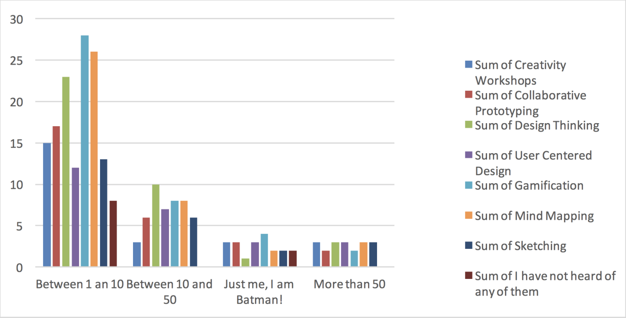


Fig. 30 Team size vs knowledge on creative approaches

Team Size vs Creative Approaches Used

With regard to the relationship between team size and the use of creative approaches, we observed that the majority of interviewees stated that they did not make use of any of the approaches described, followed by making use of "collaborative prototyping", "mind mapping" and "sketching". This is detailed in Figure 31.

Team Size vs Reception to Creative Approaches

As seen in Figure 32, we also asked the interviewees about their reception as regards learning and incorporating creative approaches into their projects. Here, we observed that small teams are, on average, more receptive than big teams.

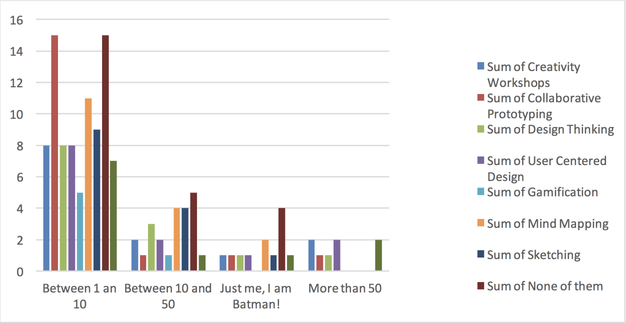


Fig. 31 Team size vs creative approaches used

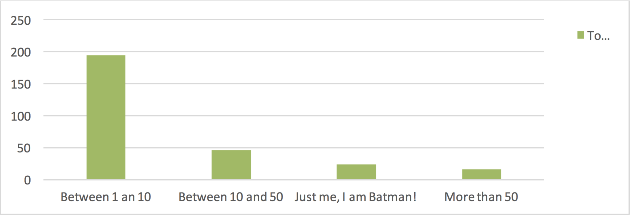


Fig. 32 Team size vs reception to creative approaches

Project Duration vs Knowledge on Creative Approaches

In the case of the relationship between the knowledge of creative approaches and the duration of a project, we did not find an outstanding link between these factors. Both short and long-term projects have similar values in terms of knowledge, as will be observed in Figure 33.

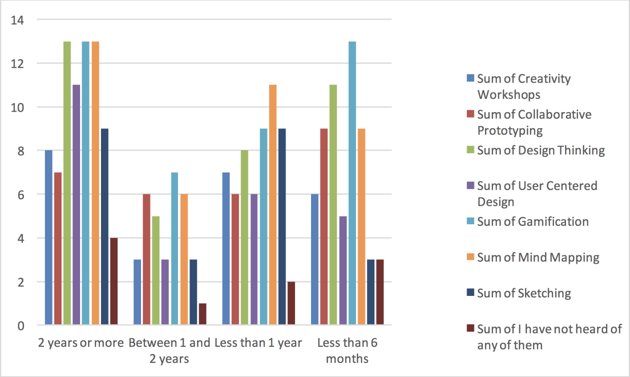


Fig. 33 Project duration vs knowledge on creative approaches

Project Duration vs Creative Approaches Used

With regard to the relationship between the use of creative approaches and the duration of a project, it would appear that short projects (with a duration of less than six months) make relatively more use of creative approaches, specifically "collaborative prototyping" and "creativity workshops". While on projects of a longer duration, not making use of creative approaches stands out as the most selected choice, in short projects, this choice moves to the third place, as shown in Figure 34.

Project Duration vs Reception to Creative Approaches

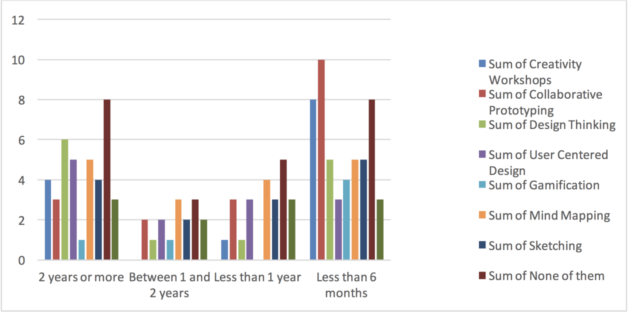


Fig. 34 Project duration vs creative approaches used

As seen in Figure 35, with regard to project duration and reception to creative approaches, it would appear that short projects tend to be, on average, more receptive to creative approaches than do longer projects.

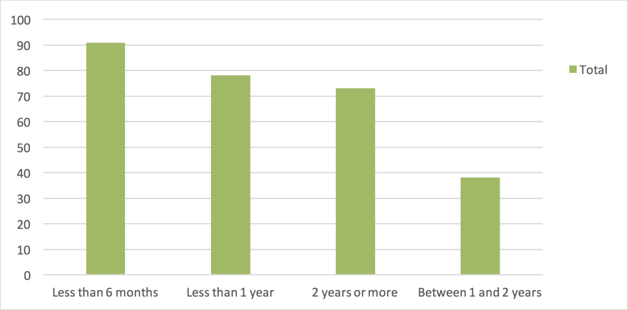


Fig. 35 Project duration vs reception to creative approaches

Years of Experience vs Knowledge on Creative Approaches

If we study the relationship between the years of experience of the professionals (Figure 36), it clearly stands out that more years of experience correspond to more knowledge of creative approaches. Within this group, the best-known approaches are: “gamification”, “mind-mapping” and “design thinking”.

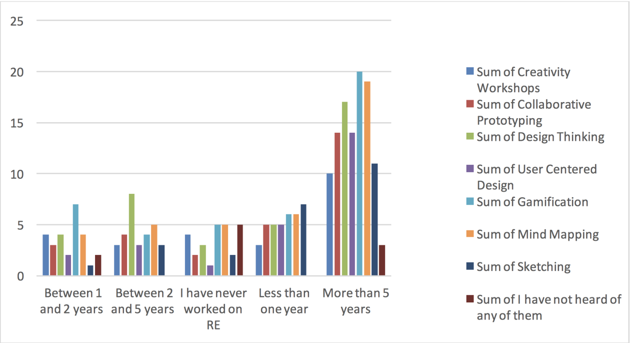


Fig. 36 Years of experience vs knowledge on creative approaches

Years of Experience vs Creative Approaches Used

Finally, in Figure 37, we analyze the relationship between years of experience in RE and the use of creative approaches. Here, it also appears that more years of experience are linked to a greater use of creative approaches.

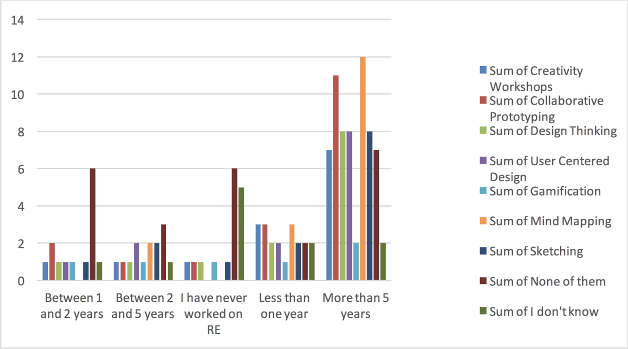


Fig. 37 Years of experience vs creative approaches used

6.2.2 Discussion of questionnaire

In the case of the team’s size, we have found that there is indeed a relationship between the size of a team and knowledge, use and reception to creative approaches. Small teams know more and are more receptive to learning new approaches. One explanation for this could be that in small teams it is easier to share knowledge owing to the reduced number of interlocutors. Collaborative prototyping, Mind mapping and sketching stand out as the most popular techniques used.

However, it seems more complicated for small teams to incorporate creative approaches into their workflows, perhaps because of their limited resources. On the other hand, teams composed of a single member do not seem to know a lot about creative approaches, which could be caused by the challenges that a single person may confront as regards being up-to-date in the latest techniques.

With regard to the duration of a project, we did not find an outstanding link between project duration and knowledge of creative approaches. However, in terms of use and reception, it would, to a slight extent, appear that short projects tend to be, on average, more receptive to creative approaches than do longer projects. One reason for this could be that, owing to the limited amount of time, short projects really need to optimize resources and make use of the best techniques to encourage creativity and be able to come up with innovative ideas.

Finally, with regard to the years of experience, the link between experience and knowledge and use of creative approaches clearly stands out. This proves to us that requirements engineering professionals are highly trained and attempt to be up to date in the latest trends in the field.

6.3 Future work

Having finalized the present study, some improvements and extensions were identified.

- *Further systematic reviews*: since this is a very new area of study, more systematic reviews are needed to identify works that have been published after this systematic review was executed. It would also be interesting to employ another combination of keywords, such as those related to innovation and user engagement.
- *Improvements related to the insights from the industry*: in order to attain a more detailed overview of the current situation in industry, a more complete questionnaire could be created. This could better analyze the specific circumstances of the professionals and projects in terms of location and resources.
- *Extension of creativity approaches to other stages of development*: we have studied creativity during the design and requirements elicitation phases. It would be interesting to investigate the employment of creativity in other areas of development, such as testing or integration.
- *Combination of proposals*: it would also be interesting to analyze a possible combination of techniques such as creativity workshops with mind mapping or sketching.
- *More empirical evidence and elaborated proposals*: since this is a very new and immature area, there is a need to apply these and other new approaches to more projects, but perhaps not necessarily to software projects only.

Acknowledgements This research has been partially funded by the Regional Government of Madrid under the SICOMORo-CM (S2013/ICE-3006) project and the ELASTIC (TIN2014-52938-C2-1-R) and MADRID (TIN2017-88557-R) projects, financed by the Spanish Ministry of Economy, Industry and Competitiveness.

References

1. Aiken, P.: Microsoft computer dictionary. Microsoft Press (2002)
2. Ambler, S.: Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons (2002)
3. Anderson, D.J.: Kanban: successful evolutionary change for your technology business. Blue Hole Press (2010)
4. Barron, F.: The disposition toward originality. The Journal of Abnormal and Social Psychology **51**(3), 478 (1955)
5. Baskerville, R.L.: Investigating information systems with action research. Communications of the AIS **2**(3es), 4 (1999)
6. Beck, K.: Extreme programming explained: embrace change. addison-wesley professional (2000)
7. Biolchini, J., Mian, P.G., Natali, A.C.C., Travassos, G.H.: Systematic review in software engineering. System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES **679**(05), 45 (2005)
8. Brown, T.: Design thinking.(m. amelang, ed.) harvard business review, 86 (6), 84–92, 141 (2008)
9. Brown, T.: Change by design. Collins Business. (2009)
10. Buzan, T.: Mind mapping. Pearson Education (2006)
11. Clark, K., Smith, R.: Unleashing the power of design thinking. Design Management Review **19**(3), 8–15 (2008)
12. Cockburn, A.: Research methods in information systems research: Matching method to researcher (2003)
13. Cockburn, A.: Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams. Pearson Education (2004)
14. Committee, I.S.C., et al.: Ieee standard glossary of software engineering terminology (ieee std 610.12-1990). los alamitos. CA: IEEE Computer Society **169** (1990)
15. Conboy, K., Wang, X., Fitzgerald, B.: Creativity in agile systems development: a literature review. In: Information Systems–Creativity and Innovation in Small and Medium-Sized Enterprises, pp. 122–134. Springer (2009)
16. Constantine, L.L.: Activity modeling: Toward a pragmatic integration of activity theory with usage-centered design. Technical Paper, Available on-line: http://www. foru se. com/articles/activitymodeling. pdf (2006)
17. Curcio, K., Navarro, T., Malucelli, A., Reinehr, S.: Requirements engineering: A systematic mapping study in agile software development. Journal of Systems and Software **139**, 32–50 (2018)
18. Davis, A., Dieste, O., Hickey, A., Juristo, N., Moreno, A.M.: Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. In: Requirements Engineering, 14th IEEE International Conference, pp. 179–188. IEEE (2006)
19. Deci, E.L., Ryan, R.M.: Self-determination. Wiley Online Library (2010)
20. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments, pp. 9–15. ACM (2011)
21. DIN, E.: 9241-210 (2011). Ergonomie der Mensch-System-Interaktion-Teil **210** (2010)
22. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and software technology **50**(9-10), 833–859 (2008)
23. Faste, R., Roth, B., Wilde, D.J.: Integrating creativity into the mechanical engineering curriculum. ASME Resource Guide to Innovation in Eng. Design (1993)
24. Franz, B.: extreme programming (2012)
25. Goguen, J.A., Linde, C.: Techniques for requirements elicitation. In: Requirements Engineering, 1993., Proceedings of IEEE International Symposium on, pp. 152–164. IEEE (1993)
26. Heikkilä, V.T., Damian, D., Lassenius, C., Paasivaara, M.: A mapping study on requirements engineering in agile software development. In: Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on, pp. 199–207. IEEE (2015)
27. Hennessey, B.A., Amabile, T.M.: The conditions of creativity. The nature of creativity **22**, 11–38 (1988)
28. Hickey, A.M., Davis, A.M.: Elicitation technique selection: how do experts do it? In: Requirements engineering conference, 2003. proceedings. 11th ieee international, pp. 169–178. IEEE (2003)
29. Hickey, A.M., Davis, A.M.: A unified model of requirements elicitation. Journal of Management Information Systems **20**(4), 65–84 (2004)
30. Higgins, J.M.: 101 creative problem solving techniques: The handbook of new ideas for business. New Management Publishing Company (1994)
31. Hollis, B., Maiden, N.: Extending agile processes with creativity techniques. IEEE software **30**(5), 78–84 (2013)
32. Hosbond, J.H., Nielsen, P.A.: Misfit or misuse? lessons from implementation of scrum in radical product innovation. In: International Conference on Agile Processes and Extreme Programming in Software Engineering, pp. 21–31. Springer (2008)

33. Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S.: A systematic literature review on agile requirements engineering practices and challenges. Computers in human behavior **51**, 915–929 (2015)
34. Jiang, L., Eberlein, A.: An analysis of the history of classical software development and agile development. In: Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on, pp. 3733–3738. IEEE (2009)
35. Jones, S., Maiden, N., Karlsen, K.: Creativity in the specification of large-scale socio-technical systems. 2007 (2007)
36. Kaufman, J.C., Beghetto, R.A.: Beyond big and little: The four c model of creativity. Review of general psychology **13**(1), 1 (2009)
37. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)
38. Kitchenham, B.A., Pfleeger, S.L.: Personal opinion surveys. In: Guide to advanced empirical software engineering, pp. 63–92. Springer (2008)
39. Kreinovich, V.: From conference papers to journal papers: Challenges and new ideas [leader's corner]. IEEE Systems, Man, and Cybernetics Magazine **2**(2), 54–55 (2016)
40. Larman, C., Basili, V.R.: Iterative and incremental developments. a brief history. Computer **36**(6), 47–56 (2003)
41. Leffingwell, D.: Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Addison-Wesley Professional (2010)
42. Lehman, M.M.: Programs, life cycles, and laws of software evolution. Proceedings of the IEEE **68**(9), 1060–1076 (1980)
43. Lehman, M.M., Ramil, J.F., Wernick, P.D., Perry, D.E., Turski, W.M.: Metrics and laws of software evolution-the nineties view. In: Software metrics symposium, 1997. proceedings., fourth international, pp. 20–32. IEEE (1997)
44. Lemos, J., Alves, C., Duboc, L., Rodrigues, G.N.: A systematic mapping study on creativity in requirements engineering. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1083–1088. ACM (2012)
45. Lindberg, T., Meinel, C., Wagner, R.: Design thinking: A fruitful concept for it development? In: Design thinking, pp. 3–18. Springer (2011)
46. Løvlie, L., Polaine, A., Reason, B.: Service design: from insight to implementation. New York: Rosenfeld Media (2013)
47. Mahaux, M., Nguyen, L., Gotel, O., Mich, L., Mavin, A., Schmid, K.: Collaborative creativity in requirements engineering: Analysis and practical advice. In: Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on, pp. 1–10. IEEE (2013)
48. Maiden, N., Gizikis, A.: Where do requirements come from? IEEE software **18**(5), 10–12 (2001)
49. Maiden, N., Jones, S., Karlsen, K., Neill, R., Zachos, K., Milne, A.: Requirements engineering as creative problem solving: A research agenda for idea finding. In: Requirements Engineering Conference (RE), 2010 18th IEEE International, pp. 57–66. IEEE (2010)
50. Maiden, N., Manning, S., Robertson, S., Greenwood, J.: Integrating creativity workshops into structured requirements processes. In: Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques, pp. 113–122. ACM (2004)
51. Maiden, N., Ncube, C., Robertson, S.: Can requirements be creative? experiences with an enhanced air space management system. In: Proceedings of the 29th international conference on Software Engineering, pp. 632–641. IEEE Computer Society (2007)
52. Maiden, N., Robertson, S.: Integrating creativity into requirements processes: Experiences with an air traffic management system. In: Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on, pp. 105–114. IEEE (2005)
53. Maiden, N.A., Rugg, G.: Acre: selecting methods for requirements acquisition. Software Engineering Journal **11**(3), 183–192 (1996)
54. Martin, R.L.: The design of business: Why design thinking is the next competitive advantage. Harvard Business Press (2009)
55. Maslow, A.H.: A theory of human motivation. Psychological review **50**(4), 370 (1943)
56. Michalko, M.: Thinkertoys: A handbook of creative-thinking techniques. Ten Speed Press (2010)
57. Nguyen, L., Shanks, G.: A framework for understanding creativity in requirements engineering. Information and software technology **51**(3), 655–662 (2009)
58. Ohno, T.: Toyota production system: beyond large-scale production. crc Press (1988)
59. Oza, N., Abrahamsson, P.: Building blocks of agile innovation. Booksurge Llc (2009)
60. Palmer, S.R., Felsing, M.: A practical guide to feature-driven development. Pearson Education (2001)
61. Pino, F.J., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. Software Quality Journal **16**(2), 237–261 (2008)
62. Plattner, H.: An introduction to design thinking process guide. the institute of design at stanford (2010)
63. Poppendieck, M., Poppendieck, T.: Lean software development: an agile toolkit. Addison-Wesley (2003)
64. Rising, L., Janoff, N.S.: The scrum software development process for small teams. IEEE software **17**(4), 26–32 (2000)
65. Rowe, P.: G., design thinking (1987)
66. Runco, M.A., Jaeger, G.J.: The standard definition of creativity. Creativity Research Journal **24**(1), 92–96 (2012)
67. Saha, S.K., Selvi, M., Büyükcan, G., Mohymen, M.: A systematic review on creativity techniques for requirements engineering. In: Informatics, Electronics & Vision (ICIEV), 2012 International Conference on, pp. 34–39. IEEE (2012)
68. Salah, D., Paige, R.F., Cairns, P.: A systematic literature review for agile development processes and user centred design integration. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering, p. 5. ACM (2014)
69. Schlosser, C., Jones, S., Maiden, N.: Using a creativity workshop to generate requirements for an event database application. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 109–122. Springer (2008)
70. Schön, E.M., Thomaschewski, J., Escalona, M.J.: Agile requirements engineering: A systematic literature review. Computer Standards & Interfaces **49**, 79–91 (2017)
71. Schuler, D., Namioka, A.: Participatory design: Principles and practices. CRC Press (1993)
72. Schwaber, K.: Agile project management with Scrum. Microsoft press (2004)
73. Schwaber, K., Beedle, M.: Agile software development with Scrum, vol. 1. Prentice Hall Upper Saddle River (2002)
74. Schwaber, K., SUTHERLAND, J.: The scrum guide–the definitive guide to scrum: The rules of the game. july, 2016. Acesso em **30** (2016)
75. Simon, H.A.: The sciences of the artificial. MIT press (1996)
76. Sliger, M.: Agile project management with Scrum. Microsoft press (2011)
77. Snider, C., Dekoninck, E., Culley, S.: Studying the appearance and effect of creativity within the the latter stages of the product development process. In: Procedings of the Second Conference on Creativity and Innovation in Design, pp. 317–328. ACM (2011)
78. Stapleton, J.: DSDM: Business focused development. Pearson Education (2003)
79. Stein, M.I.: Creativity and culture. The journal of psychology **36**(2), 311–322 (1953)
80. Sternberg, R.J., Lubart, T.I.: The concept of creativity: Prospects and paradigms. Handbook of creativity **1**, 3–15 (1999)
81. Stickdorn, M., Schneider, J., Andrews, K., Lawrence, A.: This is service design thinking: Basics, tools, cases, vol. 1. Wiley Hoboken, NJ (2011)

82. Svensson, R.B., Taghavianfar, M., Gren, L.: Creativity techniques for more creative requirements: theory vs. practice. In: Software Engineering and Advanced Applications (SEAA), 2015 41st Euromicro Conference on, pp. 104–111. IEEE (2015)
83. Takeuchi, H., Nonaka, I.: 16 the new new product development game. Japanese Business: Part 1, Classics Part 2, Japanese management Vol. 2: Part 1, Manufacturing and production Part 2, Automotive industry Vol. 3: Part 1, Banking and finance Part 2, Corporate strategy and inter-organizational relationships Vol. 4: Part 1, Japanese management overseas Part 2, Innovation and learning **64**(1), 321 (1998)
84. Tiwari, S., Rathore, S.S., Gupta, A.: Selecting requirement elicitation techniques for software projects. In: Software Engineering (CONSEG), 2012 CSI Sixth International Conference on, pp. 1–10. IEEE (2012)
85. Turk, D., France, R., Rumpe, B.: Limitations of agile software processes. arXiv preprint arXiv:1409.6600 (2014)
86. Zachos, K., Maiden, N.: Inventing requirements from software: An empirical investigation with web services. In: International Requirements Engineering, 2008. RE'08. 16th IEEE, pp. 145–154. IEEE (2008)
87. Zowghi, D., Coulin, C.: Requirements elicitation: A survey of techniques, approaches, and tools. In: Engineering and managing software requirements, pp. 19–46. Springer (2005)

Appendices

Appendix I - List of studies selected for the review (Primary Studies)

S1 Hollis, B., & Maiden, N. (2013). Extending agile processes with creativity techniques. IEEE software, 30(5), 78-84

S2 ODriscoll, K. (2016). The agile data modelling & design thinking approach to information system requirements analysis. Journal of Decision Systems, 25(sup1), 632-638.

S3 Gamble, M. T. (2016). Can metamodels link development to design intent?. In Proceedings of the 1st International Workshop on Bringing Architectural Design Thinking into Developers' Daily Activities, 14-17

S4 Newman, P., Ferrario, M. A., Simm, W., Forshaw, S., Friday, A., & Whittle, J. (2015). The role of design thinking and physical prototyping in social software engineering. 37th International Conference on Software Engineering, 2, 487-496

S5 Patton, J. (2002). Hitting the target: adding interaction design to agile software development. In OOPSLA 2002 Practitioners Reports, 1-ff

S6 Lucena, P., Braz, A., Chicoria, A., & Tizzei, L. (2016). IBM Design Thinking Software Development Framework. In Brazilian Workshop on Agile Methods, 98-109

S7 Lombriser, P., Dalpiaz, F., Lucassen, G., & Brinkkemper, S. (2016). Gamified requirements engineering: model and experimentation. In International Working Conference on Requirements Engineering,171-187

S8 Mahmud, I., & Veneziano, V. (2011). Mind-mapping: An effective technique to facilitate requirements engineering in agile software development. In Computer and Information Technology (ICCIT), 2011, 157-162

S9 Maiden, N. (2011). What Time Is It, Eccles?. IEEE software, 28(4), 84-85

S10 Wanderley, F., da Silveira, D. S., Araujo, J., & Lencastre, M. (2012). Generating feature model from creative requirements using model driven design. In Proceedings of the 16th International Software Product Line Conference, 2, 18-25

S11 Wanderley, F., & Araujo, J. (2013). Generating goal-oriented models from creative requirements using model driven engineering. In Model-Driven Requirements Engineering (MoDRE), 2013 International Workshop on,1-9

S12 Wanderley, F., Silveira, D., Araujo, J., Moreira, A., & Guerra, E. (2014). Experimental evaluation of conceptual modelling through mind maps and model driven engineering. In ICCSA 2014, 200-214

S13 Boness, K., & Harrison, R. (2007). Goal sketching: Towards agile requirements engineering. In Software Engineering Advances, ICSEA 2007, 71-71

S14 Boness, K., & Harrison, R. (2008). Goal sketching with activity diagrams. In Software Engineering Advances, 2008. ICSEA'08. The Third International Conference on, 277-283

S15 Boness, K., Harrison, R., & Liu, K. (2008). Goal sketching: An agile approach to clarifying requirements. International Journal on Advances in Software, IARIA, 1(1)

S16 Hastreiter, I., Krause, S., Schneidermeier, T., & Wolff, C. (2014). Developing UX for Collaborative Mobile Prototyping. In International Conference of Design, User Experience, and Usabilit, 104-114

S17 Sulmon, N., Derboven, J., Zaman, B., & Montero, M. (2012). Mapping Participatory Design Methods to the Cognitive Process of Creativity to Facilitate Requirements Engineering. Information Systems Research and Exploring Social Artefacts: Approaches and Methodologies, 221-241