Creativity Patterns Guide: Support for the Application of Creativity Techniques in Requirements Engineering

Elton R. Vieira[1], Carina Alves[1], and Letícia Duboc[2]

[1] Informatics Center, Federal University of Pernambuco
[2] Computer Department, State University of Rio de Janeiro
{ervs,cfa}@cin.ufpe.br, leticia@ime.uerj.br

Abstract. Creativity techniques are tools for stimulating creative thinking. The importance of creativity fostering techniques in software development has been recognized and investigated by researchers for over a decade, yet the greater software engineering (SE) community makes little use of the myriads of techniques available. In order to encourage a wider adoption and to support the use of creativity techniques in software development, we have reviewed a large number of creativity techniques and have created a Creativity Patterns Guide. This paper describes the part of the guide tailored for the requirements engineering phase. The guide has been evaluated in real-world projects.

Keywords: Creativity Techniques, Requirements Engineering, Design Patterns.

1 Introduction

Software development is a problem-solving activity, from software inception to testing. Many of these can be solved creatively. Creativity is particularly important in the requirements engineering (RE) phase. The obvious case is market-driven products, which must contain innovative features for attracting customers [1]. Furthermore, the requirements analyst is often faced with problems involving different disciplines, multiple stakeholders, a highly dynamic context, uncertainties and trade-offs. Understanding and solving such complex problems also requires creativity [2]. Research suggests that problems can be solved creatively through the use of techniques that stimulate idea generation [3, 4]. Some techniques have been studied in the RE context and shown to be powerful tools for elaborating requirements [2, 7]. Despite of benefits and the myriads of techniques available [9], the software engineering community still makes a limited use of creativity techniques [8].

The contribution of this work is two-folded. We have reviewed a large number of creativity techniques, selecting the ones that we believe to be suitable for stimulating creativity in software development. We have also adapted and organized them with a structure, language and examples that are familiar to software developers. This paper presents the Creativity Patterns Guide, a catalogue of creativity techniques for supporting idea generation during software development. In order to facilitate its adoption by software engineers, creativity techniques are presented following the familiar concept of Design Patterns [16]. The reminder of the paper is organized as

M. Winckler, P. Forbrig, and R. Bernhaupt (Eds.): HCSE 2012, LNCS 7623, pp. 283–290, 2012.

follows. Section 2 discusses related work. Section 3 describes the steps to develop the Creativity Patterns Guide. Section 4 presents the results of empirical studies analyzed using the Goal-Question-Metric (GQM) technique [21]. Finally, Section 6 summaries the main contributions and briefly discusses future work.

2 Background

Even though there is no consensus on a definition of "creativity", researchers seem to agree that the term is related to the identification and resolution of problems [11]. It is also accepted that creativity is not necessarily a natural-born talent, but something that can be encouraged, supported and trained [10]. Creativity techniques are tools that can stimulate creative thinking and idea generation [3] from a rich variety of activities [4]. The literature presents a large number of such techniques [5, 6, 9].

In the software industry, the development of innovative market-driven products and services has intensified the discussion about creativity [1]. Software development is a problem-solving activity, from start to end, and could therefore benefit from techniques that help problems to be solved creatively. Yet, little space is dedicated for practices or the study of techniques that stimulate creativity in the software development process [15]. A notable exception is the field of requirements engineering, in which a few research groups have been studying creativity for over a decade [16]. Even though creativity is recognized as an important aspect of software development, and requirements engineering in particular, only a handful of techniques have been studied and the wider software engineering community still pays little attention to such techniques [8]. With this goal in mind, we have built the Creativity Patterns Guide, following the structure of design patterns.

Gamma et al. [17] created patterns that described common problems in software design, generic solutions to these problems and the consequences of using them. They follow the same structure, which eases the understanding and application of the pattern [17]. Design patterns are not mutually exclusive; in fact they are rarely used separately and are best combined [18]. They can also be categorized by the activity they support. Design patterns have been widely used by software developers [18]. Like design patterns, creativity techniques are best applied in combination [19]. They also can be classified along a number of categories that will help to pick and choose the best technique for a particular context. Following the idea of design patterns, we have created a catalogue where creativity techniques are explained, categorized and exemplified in the context of software engineering.

3 The Creativity Patterns Guide

One of the main challenges for the adoption of creativity techniques is to identify which techniques to use on a particular context [9]. Both creativity techniques and the phases of the software development have particular characteristics. Our guide categorizes creativity techniques and maps them to the software development phases, so that engineers can pick the most appropriate ones. In this paper, we report on the part of the catalogue concerning the *requirements engineering (RE)* phase; techniques for this phase were selected as follows:

Step 1 involved an ad-hoc search of digital libraries to identify articles and books discussing the topic "creativity" [5, 6, 9, 10], which led to a list of 254 techniques.

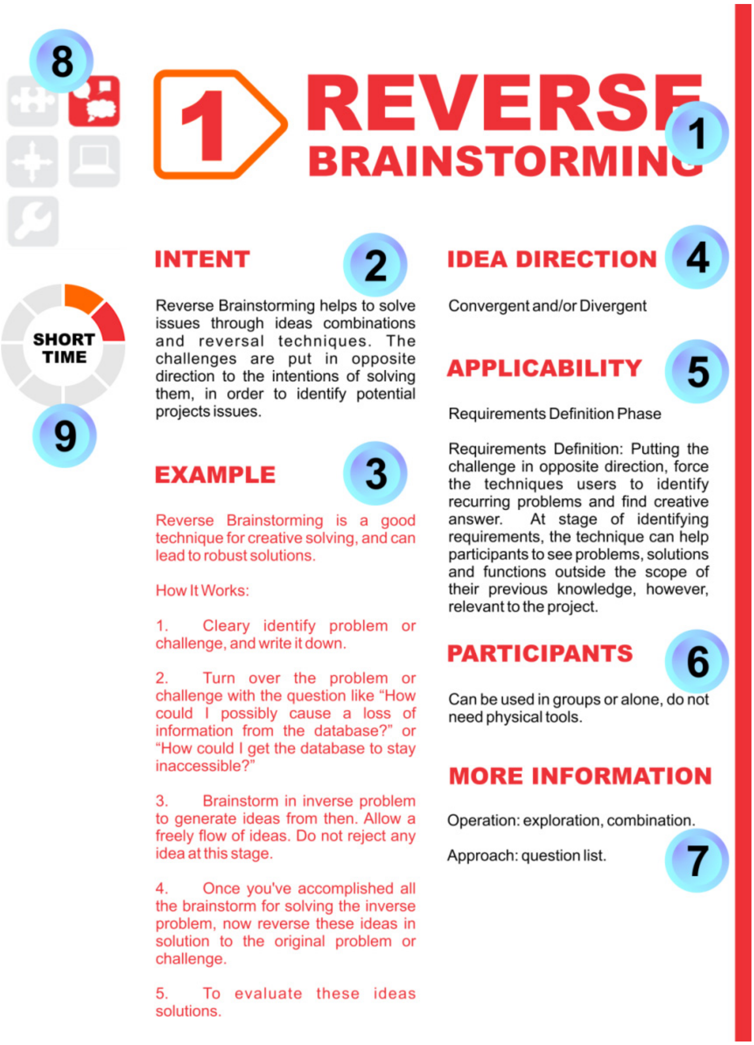
Step 2 analyzed the objectives and operations of each technique, and the ones considered redundant or too similar were removed, reducing the list to 98 techniques.

Step 3 evaluated the techniques with respect to its suitability to the RE phase. First we selected techniques that could be adapted to the software context, that is, techniques designed to specific areas, such as arts or marketing were removed. Second, we have classified them according to four criteria proposed in the creativity literature and then examined the RE literature to map this criteria to RE activities.

The four criteria for classifying the techniques were: (i) The control criteria, defined by Grube and Schmid [14] describes whether the technique requires physical artifacts or not, and whether it is applicable to a group or an individual. (ii) The operation criteria classify the techniques according to the four model types defined by Boden [13]: exploration, combination, transformation and evaluation [9]. (iii) The approach criteria, also defined by Schmid e Grube [9], groups techniques according to their basic activity: free association, different viewpoints, questions list, structuring, area expert and random entry [14]. (iv) Finally, the direction criteria determine whether the style of the technique is convergent or divergent [13].

RE comprises the activities of: identification of stakeholders, recognition of different viewpoints, problem understanding, scope definition, and elaboration, negotiation, documentation and validation of requirements [20]. In order to map these activities to the above criteria, we reviewed the literature, looking for observations and studies that suggested particular mappings. For example, Pressman [20] observes that requirements may be elicited by means of group meetings or individual interviews [8], which suggests the suitability of techniques classified as "individual" or "group. Kirton observes that requirements analysis demands convergent thinking to validate requirements [12], indicating that a technique classified as "convergent" might be suitable for the requirements validation activity. As a result of this mapping, 41 creativity techniques were judged suitable for the RE phase. The list of techniques and their mapping can not be shown for space constraints. While we sought to ground our mapping on well-established research, we recognize that this mapping is subject to our interpretation of the works reviewed. Further validation will assess its appropriateness.

The **Creativity Patterns Guide** was developed to encourage and support the use of creativity techniques in the context of software development. For such, we have taken the following measures: (i) adopted the familiar structure of design patterns, (ii) illustrated the techniques with software-related scenarios, (iii) adapted the language to software engineers, and (iv) used an attractive layout with decentralized columns, symbols and highlighted concepts. As show in Figure 1, each technique is described by a short and expressive (1) **name**, its (2) **intent**, an (3) **example** in the software context, an (4) **idea direction** (convergent or divergent), its (5) **applicability** (through scenarios), its (6) **participants** characteristics (individual, group, resource), (7) **more information**, a (8) **symbol** representing the software development phase, and the (9) **time** it requires.


8
1
9
2
3
4
5
6
7
SHORT TIME

1 REVERSE BRAINSTORMING

INTENT

Reverse Brainstorming helps to solve issues through ideas combinations and reversal techniques. The challenges are put in opposite direction to the intentions of solving them, in order to identify potential projects issues.

EXAMPLE

Reverse Brainstorming is a good technique for creative solving, and can lead to robust solutions.

How It Works:

1. Cleary identify problem or challenge, and write it down.

2. Turn over the problem or challenge with the question like “How could I possibly cause a loss of information from the database?” or “How could I get the database to stay inaccessible?”

3. Brainstorm in inverse problem to generate ideas from then. Allow a freely flow of ideas. Do not reject any idea at this stage.

4. Once you've accomplished all the brainstorm for solving the inverse problem, now reverse these ideas in solution to the original problem or challenge.

5. To evaluate these ideas solutions.

IDEA DIRECTION

Convergent and/or Divergent

APPLICABILITY

Requirements Definition Phase

Requirements Definition: Putting the challenge in opposite direction, force the techniques users to identify recurring problems and find creative answer. At stage of identifying requirements, the technique can help participants to see problems, solutions and functions outside the scope of their previous knowledge, however, relevant to the project.

PARTICIPANTS

Can be used in groups or alone, do not need physical tools.

MORE INFORMATION

Operation: exploration, combination.

Approach: question list.

Fig. 1. Elements of Creativity Patterns Guide

4 Empirical Study

An empirical study was conducted in two software development projects to analyze the impact of using the Creativity Patterns Guide during the Requirements Engineering phase: **Project A** involved the development of Advergame. The team comprised 6 professionals: one project manager, one requirements analyst one senior developer, two junior developers and one designer. The project lasted two months. **Project B** consisted on the development of Educational Networking applications. The team had 8 professionals: one project manager, one designer, one specialist in social networks one senior developer and four junior developers. This team has expertise in web development, Human-Computer Interaction, testing, social media and educational software development. The project has been developed for the last 3 years and it is still in execution. The study was conducted in the installations of each company. One of the authors acted as facilitator.

The main objective of the empirical study was evaluating the overall benefits of our proposal. The Goal Question Metric approach [21] was adopted to structure the study. The GQM is a technique that helps to define metrics for particular goals. It starts with a goal, define questions for assessing goals and create metrics that allows for questions to be answered. The main goals, questions and metrics are described in Table 1. Goal 1 evaluates the overall acceptance of the guide, while Goal 2 evaluates number of ideas and requirements that are considered relevant to the final product. This is a similar criteria used to evaluate other methods for stimulating creativity in RE, such as [7, 9, 10, 15].

Table 1. Goals, Questions and Metrics of the GQM Technique

Goals	Questions	Metrics
(G1)) Assess the acceptance of the Creativity Patterns Guide by development teams as a suitable tool to stimulate creativity.	(G1Q1) Participants considered the guide ease to use?	Opinions regarding the usability of the guide. Suggestions to improve the guide.
	(G1Q2) Participants would use the guide in other projects?	Opinion regarding their interest to adopt the guide in future projects.
	(G1Q3) The structure of the guide based on design patterns facilitates its use?	Comments regarding the influence of design patterns to explain creativity techniques. Data regarding previous knowledge and experience of participants in design patterns.
(G2) Assess the usefulness of the Guide to help development teams generating ideas and requirements.	(G2Q1) How many idea alternatives were generated?	Number of ideas generated during the study.
	(G2Q2) The idea alternatives generated may be adopted to develop the final product?	Participant opinion on the relevance of generated ideas. Number of ideas that can be included as real requirements.

Each team received a fragment of the guide, containing techniques that were used during a creativity workshop. Initially, the facilitator explained the guide and gave the techniques presented in the Guide as depicted in Figure 1. Project A used the Method 635 and Reverse Brainstorming creativity techniques. While Project B used the techniques Method 635, Reverse Brainstorming, Free Association and Provocation. The techniques were selected because they were considered fast to apply, yet possessing great creative potential. The ideas generated were written in whiteboards to be discussed by all team members. The study with Project A had one workshop session that lasted 2 hours, and the Project B study had one session that lasted 4 hours. After the workshops, we conducted a short focus group to discuss their impressions of the Guide. In addition, we also collected data by applying a semi-structured interview. Eight participants were interviewed, two from Project A and six from Project B. Each interview lasted less of 30 minutes. There were three basic questions about the Guide (related to G1) and two questions about the quality of ideas (related to G2).

4.1 Results

The results of the study are classified to answer the questions stated in the GQM technique as presented in the previous section:

(G1Q1) – Participants provided positive impressions after using the Guide. They stated that techniques are well explained and easier to apply in the context of software development. All participants affirmed they understood the techniques and concepts presented in the Guide, even those who had never used creativity techniques. The visual design and contextualization of creativity techniques for software development scenarios were considered important features. In terms of criticism, participants posed that the Guide still needs to improve the description of tasks and some examples for using techniques. Participants also proposed the following suggestions: separate the techniques in blocks, specifying which SE phase that block is intended; specifying the amount of time required to apply each technique. These suggestions were incorporated in the guide as presented in the paper.

(G1Q2) – All team members of both projects confirmed they would use the guide to stimulate creative thinking in other projects. Even though, these impressions are considered a rather subjective measure, we believe that these preliminary positive feedbacks indicate the guide is a promising approach to integrate creativity activities with other traditional phases of SE. Participants also stated the guide was fun the use and could be a relaxation moment to escape from the constant pressure of projects.

(G1Q3) – The adaptation of creativity techniques to the context of RE, and the use of concepts already known in design patterns, directly influenced the rapid assimilation of the Guide by participants. These characteristics received a unanimous positive response. The creativity activities conducted by the participants of the study were guided only by the examples present in the technique description. The majority of participants had previous contact with design patterns, this familiarity led to a direct understanding of elements present in the Creativity Patterns Guide.

(G2Q1) – The total of ideas generated was 121. In Project A, participants generated 37 different ideas; while in Project B, participants proposed 94 ideas. It is worth noting that not all idea alternatives were incorporated as requirements. Another aspect to analyze is the substantial difference between the number of ideas generated by both teams. This may be due to the fact that study with Project B lasted 4 hours.

(G2Q2) – The number of ideas generated was considered high by the teams. In Project A, more than 70% of ideas had direct influence in the requirements for the advergame. In Project B, we could not assess the real adoption of ideas generated because the project was not fully implemented by the time of writing this paper. Regarding participants opinion on the relevance of generated ideas, they observed that some ideas were useful triggers to develop more innovative features. These opinions were used to measure the quality of ideas.

As the subjects selected few techniques, the usefulness of the guide for stimulating creativity could only be assessed for the selected techniques. In order to assess the full guide, more experiments are needed. It is also important to note that this study did not aim to evaluate the completeness of the guide or whether the structure of the guide has helped subjects to select the best techniques for the problem in hand. Nevertheless the volunteers were asked about the suitability and ease of use of the guide for understanding and selecting techniques. The results in terms of number of ideas and their impact in the final product showed that the chosen techniques were appropriate to stimulating creativity.

5 Conclusions and Future Work

This paper presented the Creativity Patterns Guide, an artifact to stimulate the adoption of creativity techniques in software engineering. The guide adopts the familiar concepts of Design Patterns to explain the use of creativity techniques in the context of software development. To construct the guide we gathered relevant knowledge on creativity techniques available in literature, mapped this knowledge using four classification criteria and structured this knowledge in a catalogue as elements of patterns. Our aim to develop the Creativity Patterns Guide was to provide a practical and ease to use tool to enhance creative thinking of development teams. In particular, a key contribution of our approach is to encapsulate a relevant knowledge body in creativity using appropriate vocabulary and providing several examples understandable for software engineers.

In order to conduct an initial evaluation of the guide, we employed the GQM technique to structure the goals of the study and help collecting empirical data to answer the proposed questions. As key results, we observed that the analogies with design patterns were considered a very positive manner to structure creativity knowledge. These features enabled the easy learning and application of presented creativity techniques. Even participants who had never used creativity techniques confirmed the simplicity of the guide. It is important to observe that the adoption of design pattern does not aim to propose a formal pattern to adopt creativity. In fact, the patterns act more as a guidance to use existing creativity techniques. It serves as a basic reference for software engineers to adopt creative thinking more easily in their daily activities. In addition, given that creativity is a very subjective and intuitive task, it is not possible to assess if participants selected a correct or wrong technique. Further research is needed to assess the overall suitability of the guide. We obtained very positive responses regarding the acceptance of the guide. However, in order to obtain representative and reliable data we need to conduct more formal studies. Finally, we aim to perform experiments comparing the number and quality of ideas generated by two teams, one using the guide and the other team applying creativity techniques published in other sources.

References

1. Alves, C., Ramalho, G., Damasceno, A.: Challenges in Requirements Engineering for Mobile Games Development: The Meantime Case Study. In: Requirements Engineering Conference (2007)

2. Solís, C., Ali, N.: Distributed Requirements Elicitation Using a Spatial Hypertext Wiki. In: IEEE International Conference on Global Software Engineering (2010)
3. Forster, F., Brocco, M.: Understanding Creativity-Technique Based Problem Solving Processes. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 806–813. Springer, Heidelberg (2008)
4. Norman, D.A.: Emotional Design: Why do We Love (Or Hate) Everyday Things. Basic Books (2004)
5. Mycoted (2012), http://www.mycoted.com/
6. Mind Tools (2012), http://www.mindtools.com/
7. Maiden, N., Jones, S.: Provoking Creative Design: Making it Scale. British Computer Society (2008)
8. Maiden, N., Jones, S., Karlsen, K., Neill, R., Milne, A.: Requirements Engineering as Creative Problem Solving: A Research Agenda for Idea Finding. In: 18th IEEE International Requirements Engineering Conference (RE), Sydney (2010)
9. Grube, P., Schmid, K.: Selecting Creativity Techniques for Innovative Requirements Engineering. In: 3rd International Workshop on Multimedia and Enjoyable Requirements Engineering (2008)
10. Michalko, M.: Thinkertoys. A handbook of creative-thinking techniques, 2nd edn. TenSpeed Press, Berkley (2006)
11. Mich, L., Franch, M., Berry, D.M.: Classifying Web-Application Requirement Ideas Generated Using Creativity Fostering Techniques According to a Quality Model for Web Applications. In: Proceedings of the 12th International Workshop on Requirements Engineering: Foundation for Software Quality, Luxembourg (2006)
12. Kirton, M.J.: Adaptors and innovators: Styles of creativity and problem solving. Routledge, London (1989)
13. Boden, M.A.: Computer models of creativity. AI Magazine 30(3), 23–34 (2009)
14. Grube, P., Schmid, K.: State of Art in Tools for Creativity. idSpace Project. University of the Netherlands (2008)
15. Glass, R.L.: Software Creativity 2.0. Developer *Books (2006); Mich, L., Anesi, C., Berry, D.M.: Requirements Engineering and Creativity: An Innovative Approach Based on a Model of the Pragmatics of Communication. In: Proceedings REFSQQ Workshop (2004)
16. Lemos, J., Alves, C., Duboc, L., Rodrigues, G.: A Systematic Mapping Study on Creativity in Requirements Engineering. In: ACM Symposium of Applied Computing (2012)
17. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading (1995)
18. Braz, S.: A Qualitative Assessment of Modularity in CaesarJ components based on Implementations of Design Patterns. M.Sc. thesis, Universidade Nova de Lisboa (2009)
19. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, E.: Pattern-Oriented Software Architecture: A System of Patterns, vol. I. John Wiley & Sons, England (1996)
20. Pressman, R.: Software Engineering, 6th edn. Mcgraw Hill (2006)
21. Basili, V., Caldeira, G., Rombach, H.D.: The Goal Question Metric Approach. In: Marciniak, J. (ed.) Encyclopedia of Software Engineering. John Wiley & Sons, USA (1994)