

Does Software Speak Creativity? Creative Processes and Techniques in the Design of Software Applications

Claudia Iacob

Department of Computer Science, University of Milan

iacob@dico.unimi.it

ABSTRACT

This paper investigates the use of 3 creative techniques - namely scenario-based design, sketching, and mockup creation - in software design processes. During a series of design workshops, 8 teams of designers were asked to design the GUI and the interaction process for software applications for domains such as collaborative drawing, collaborative text editing, and collaborative games. Each workshop was organized following the model of definition for a creative process as proposed by Graham Wallas and for each phase of the process a different technique was used by the participants. The paper describes the 3 techniques in the context of a creative process, their use during the workshops, and a brief discussion on their impact on the overall design processes followed by the teams.

1. INTRODUCTION

Software is rarely associated with creativity. However, software design, as any type of design, is a highly creative endeavor [1]. It implies the key steps of any design process - problem finding and problem solving, understanding and defining problems, balancing forces and coming up with creative solutions. For that reason, it could highly benefit from the techniques and concepts revolving around creativity.

This paper describes the investigation of the use of 3 creative techniques - scenario-based design [2], sketches [3], and mockups [4] - in the design of the interaction and the graphical user interface (GUI) of software applications. Scenario-based design is defined as the technique of using scenarios during design processes. In terms of software design, scenarios should answer questions such as: who are the users?, what are their goals?, what is their motivation to use the software application?, how could they use the application, and when and where can the application be used?. Sketches are tools for capturing preliminary observations and ideas. They are widely used in various domains, including graphic design, and architecture. Mockups are early prototypes made of cardboard or otherwise low-fidelity materials. During a series of design workshops, 8 teams of designers were asked to design the GUI and the interaction process for software applications for a specific collaborative domain. Each workshop was organized following the model of definition for a creative process as proposed by Graham Wallas [5] and for each phase of the process a different technique was used by the participants.

2. CREATIVE PROCESSES AND TECHNIQUES

Creativity has been defined in terms of several concepts, two of those being the creative process, and the techniques for creativity.

2.1 Creative Processes

A first attempt of a creative process model definition was proposed in 1926 by Graham Wallas [5] who identifies four

phases in a creative process: i). preparation as the phase in which the problem to be solved is clarified and understood, ii). incubation when one no longer consciously considers the problem, iii). illumination as the phase in which the creative insight occurs, and iv). verification, the last phase during which it is verified that the creative insight is indeed a solution for the problem to be solved. Osborn [6] refines this definition and proposes a two-phase model for defining a creative process. The first phase consists of the idea generation, being followed by a second phase called idea evaluation. Another similar model was proposed by Amabile [7] which defines the creative process based on four phases: i). problem presentation, ii). preparation, iii). response generation, and iv). response verification.

2.2 Creative Techniques

There are several creative techniques documented in the literature; of interest to this work are some of those mostly fitting the context of software design.

2.2.1 Scenario-based Design

The complexity of software design problems exceeds one's individual ability to tackle them and asks for the collaboration of stakeholders with different expertise and backgrounds. In addition to that, it is often the case that software addresses the problems of clients (often, the users) which may not be (and are not willing to be) experts in software design. Therefore, one of the challenges in designing software applications is finding ways to communicate design ideas and interaction representations. One way to do that is by describing as in a story the actors (the potential users of the application) and the activities (the actions the application supports) they would perform when engaged in the interaction with the application under design. Such descriptions are called scenarios. They are stories which describe people in action, their goals, and motivation, the concrete descriptions of activities that engage the user when performing a specific task. Scenario-based design - i.e. the technique of using scenarios during design processes - has been applied in various stages of the software development cycle, such as requirements analysis, user-designer communication, design rationale, documentation and training, evaluation, abstraction and team building [2].

2.2.2 Sketches

In addition to supporting externalization processes [3], sketches have been documented to enhance social interaction, coordination among collaborating designers [8], and introspection [9]. The initial phases of GUI design are governed by uncertainty. Objects belonging to the GUI may have "uncertain types, sizes, shapes, and positions" [20]. It is because of this uncertainty that designers do not feel the burden of deciding on details such as colors, alignments, and fonts which should be decided on during later phases of the process. Sketching GUIs gives the process certain fluidity, since sketches are rough representations and may be

modified at any time based on the continuous exploration of the design space of the application.

2.2.3 Mockups

Mockups are early prototypes made of cardboard or otherwise low-fidelity materials. They resemble the final product, but only at a surface level, having little of the eventual functionality [4]. Mockups help designers negotiate on UI design related aspects and may be easily created by anyone involved in the process. Several tools exist for creating mockups, the most popular being Balsamiq [10] and Mockingbird [11].

3. CREATIVITY IN SOFTWARE DESIGN

The section reports on the application of the presented creative techniques during 8 design workshops which were organized following the model of definition of a creative process as proposed in [5]. During a design workshop, a team of 3-5 designers were asked to design the GUI and the interaction process of a software application for a specific problem domain from the list: collaborative drawing, collaborative text editing, collaborative puzzle solving, and collaborative crosswords solving. Each workshop had 3 phases – preparation, incubation, and illumination - and the 31 participating designers were encouraged to use different creative techniques during each phase.

3.1 Creative Processes in Software Design

3.1.1 Preparation

During the preparation phase, the teams were asked to choose a problem from the list and to generate as many scenarios ideas for an application to tackle this problem. 143 ideas were generated which is an indicator of the fact that the teams found the design space exploration not only useful for finding possible solutions, but also a means of understanding the other members' backgrounds and expertise.

3.1.2 Incubation

During the incubation phase, the teams were asked to choose another problem from the list and find as many differences and similarities as possible between this problem and the one chosen during the previous phase. The total number of ideas generated was 110, out of which 44 were similarities between 2 different problem spaces and 66 were differences. As a consequence, the participants were supported in understanding the secondary implications in their design processes and to get a different perspective.

3.1.3 Illumination

The last phase required participants to sketch the GUI and the interaction process they agreed on for an application to tackle the problem they initially chose. During the illumination phase, the teams went back to the ideas previously generated. They sketched a subset of these ideas in order to get insight and inspiration. Their sketching process was not linear, all teams moving back to previous representations, modifying or combining them in order to create new representations and components. The use of sketches is discussed further on in the next section.

3.2 Creative Techniques in Software Design

3.2.1 Scenario-based Design

Through the use of scenario-based design technique, the total number of ideas generated by the 8 teams was 143. 38% of these ideas addressed aspects related to the functionality of the application under design. 28% of the scenario ideas related to the

goals the users could have when interacting with the application under design. Exploring the potential users the application might have generated 14% of the ideas. 12% of the ideas the participants generated addressed the possible motivation for using the collaborative application. The large number of ideas generated and their diversity supported the exploration of a broader number of design situations and possible good practices to tackle them.

3.2.2 Sketches and Mockups

Sketches were created by all the teams. These representations included the GUI and elements of interaction design. They were used mainly for expressing design ideas and included the design of controls, metaphors, errors, interaction means, input and output channels, users in action, features, layouts, and appearances. Sketches also served as communication means, supporting the members of each team in understanding each others' ideas and adding to them. This reflected in the sketches of mind maps, intermediary representations of the concepts they wanted to include in the design, and explanatory diagrams of concepts and the relationships existing between them.

4. CONCLUSIONS

The paper reports on an investigation of the use of creative techniques such as scenario-based design, sketches, and mockups in software design processes. 8 teams worked on the design of software applications addressing one of the problems: collaborative drawing, collaborative text editing, collaborative puzzle solving, and collaborative crosswords solving. Their design processes followed 3 phases [5], a creative technique being used during each phase.

5. ACKNOWLEDGMENTS

Special thanks to all the participants in the workshops for the inspiring ideas and great designs.

6. REFERENCES

- [1] Mangano, N., Baker, A., Dempsey, M., Navarro, E., van der Hoek, A. 2010. Software design sketching with calico. *ASE '10*. ACM, New York, NY, USA, 23-32.
- [2] Carroll, J. M. 1995. *Scenario-Based Design: Envisioning Work and Technology in System Development*. John Wiley & Sons, Inc., New York, NY, USA.
- [3] Landay, J. A., Myers, B. A. 2001. Sketching Interfaces: Toward More Human Interface Design. *Computer* 34, 3 (March 2001), 56-64.
- [4] Gregory, S. T.. 1984. On *prototypes* vs. *mockups*. *SIGSOFT Softw. Eng. Notes* 9, 5 (October 1984), 13-13.
- [5] Wallas, G. 1926. *The Art of Thought*. New York, Harcourt, Brace and Company
- [6] Osborn, A. 1953. *Applied Imagination: Principles and Procedures of Creative Problem Solving*. New York, New York: Charles Scribner's Sons
- [7] Amabile, T.M. 1983. *The Social Psychology of Creativity*. Springer-Verlag, New York
- [8] Baskinger, M. 2008. Pencils before pixels: a primer in hand-generated sketching. *Interactions*, 15, 2 (Mar. 2008), 28-36.
- [9] Schön, D. 1983. *The reflective practitioner: How professionals think in action*. NY: Basic Books
- [10] Balsamiq, <http://balsamiq.com/>
- [11] Mockingbird, <https://gomockingbird.com/>