Check for updates

Sketching by Cross-Surface Collaboration

Jorge Luis Pérez-Medina[1(✉)] and Jean Vanderdonckt[2]

[1] Intelligent and Interactive Systems Lab (SI^2 Lab), Universidad de Las Américas (UDLA), Quito, Ecuador
jorge.perez.medina@udla.edu.ec
[2] Université catholique de Louvain (UCL), LouRIM - Place des Doyens, 1, 1348 Louvain-la-Neuve, Belgium
jean.vanderdonckt@uclouvain.be

Abstract. Human-Computer Interaction and agile practices in software engineering are not two separate domains, but rather agile is a work principle that is applied in software development, of which user interface design represents a significant part. Agile projects could require such an approach, which typically iterates on the user interface using low-fidelity prototypes. This paper motivates, presents, and assess capabilities of a software for collaborative sketching of user interfaces on multiple surfaces of interaction, ranging from mobile phones to wall screens. We proposes a Collaborative User Centered Design (CUCD) method for user interface prototyping to supporting cross-surface collaboration by sketching, enabling fast, flexible, intuitive and reusable prototype.

Keywords: Collaborative Design · Electronic sketching · Multi-surface · Prototyping · User interface design

1 Introduction

While software modeling tools are more often used during early steps of the software development lifecycle (SDLC) [16], including the User Interface (UI) [5], other techniques are followed such as: free-form approaches. Office tools, white boards and papers arranged on a wall, are frequently used [20]. Indeed, these can help to produce more and usable tools and bring together people who understand the activities and needs of developers and other stakeholders to produce User Interface Design (UID) throughout the SDLC.

Designers can explore the requirements using essential UI prototypes (UIP) [1]. It represents UI requirements in a technology-independent manner. Although essential UIP many stakeholders find them too abstract and instead prefer screen sketches and not surprisingly working software [1]. Designers will use the simplest tool which will get the job done. Whiteboard diagrams typically suffice for initial requirements models. The Agile Software development (ASD) encompasses a series of methods for software development. A survey [1] concludes that 92.7% of respondents indicated that agile teams were using *Whiteboard Sketching*

© Springer Nature Switzerland AG 2019
Á. Rocha et al. (Eds.): ICITS 2019, AISC 918, pp. 386–397, 2019.
https://doi.org/10.1007/978-3-030-11890-7_38

for requirements gathering. 85.5% of those teams found the effort worthwhile. Similarly, sketching activities were the most common primary approach to modeling [2]. Over 90% of respondents [3] indicated that their agile teams did some sort of up front requirements modeling. These results are aligned to the daily effort taken by User Experience (UX) designers during gathering requirements. It consists of using sketching in early of the UI design [9] as an informal and flexible manner for producing any models for requirements and analysis phases. Sketching can also be used by any stakeholders as it is a way of expression which does not require any advanced modelling skills. Sketches are therefore refined to achieve user validation.

We aim at providing support for collaborative and flexible modeling in early steps of a User Centered Design (UCD) process where stakeholders and final users can collaborate by using multi-surface provided by multi-device. It consists of using sketching in early of the UI design. Sketching can also be used by any stakeholders using any device as it is a way of expression which does not require any advanced modelling skills. Sketches are therefore refined to achieve user validation. Iterative UID tools, used in Human-Computer Interaction (HCI), generally support electronic sketching, giving more freedom to change sketches and more flexibility in creating and evaluating a design prototype [10]. Prototyping the UI can detect ergonomic issues that will arise even before the first line of code is produced. The prototype is also a valuable communication tool; the

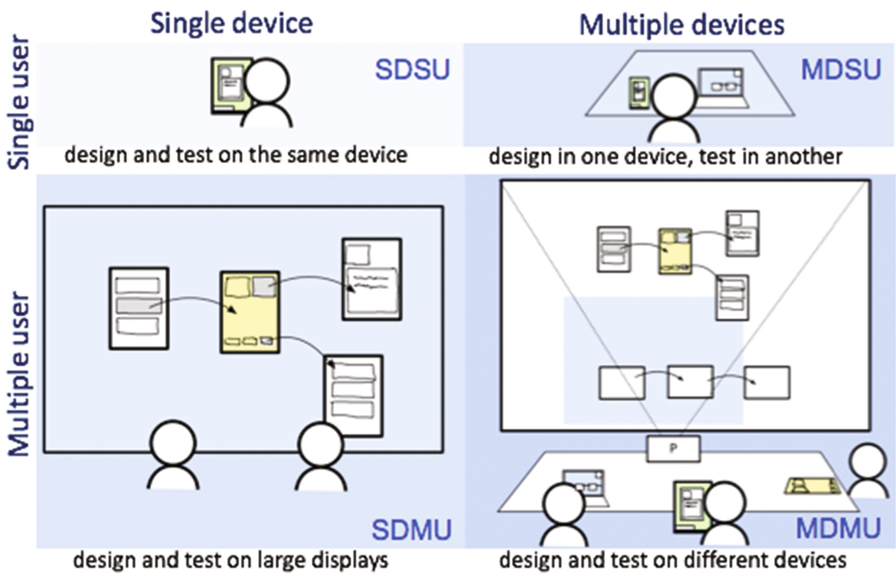

Fig. 1. Four physical configurations of a CUCD method considering multiple users and devices: SDSU = single device and single user; SDMU = single device and multiple users; MDSU = multiple devices and single user; MDSD = multiple devices and multiple users.

operation of a tool can be described to a non-computer user with greater ease. Prototyping is characterized by collaborative interactions, therefore a tool to support UID should ideally be flexible in order to accommodate different design situations, i.e., different activities, people from different areas using different media and devices.

We propose a combination of four possible configurations (Fig. 1) supporting collocated synchronous interactions identified in [15]. It considers the different users (such as, designers and users) and devices. This paper presents a flexible CUCD method based on sketching to support UI prototyping along a SDLC. It promotes flexible communication among stakeholders and end users in early steps of development when it is vital to understand the system requirements. Section 2 presents some related work including sketching and basic concepts of design by sketching. Section 3 describes the CUCD method. Section 4 introduces the Collaborative Sketching System (CSS) which supports the enactment of the CUCD method. An experimental report concludes this section. Finally, conclusions and some remarks for future work are presented.

2 Related Work

Sketches [6] are rapid drawings. Stakeholders and final users are able to carry out UID by sketching during collaborative sessions of brainstorming for requirements gathering. Sketching is useful as a design tool. It stimulates a re-interpretive cycle in the individual designer's idea generation process [12]. The prototyping activity is then complementary to the sketching activity in order to achieve understanding, especially if the artifact being designed has more dimensions than those used by the designer while sketching. Design is a hard challenging for designers. During the initial phases of this process designers have only a vague goal in mind. They does not have a clear understanding of what to produce. Design is essentially a problem of wicked nature[1]. On wicked problem the complexity can be due incomplete or contradictory knowledge, the number of people and opinions involved, the large economic burden, and the interconnected nature of these problems with other problems. In the HCI domain, UID shares the same aforementioned properties: the wicked nature, the use of *"cheap"* tools like sketching, the use of low-fidelity models such as prototypes, etc. The main difference is that UID is included in more general methods such as UCD which has specific roles with different responsibilities to the party involved.

Prototyping is one the core activities of UCD process. A review of past and current graphical user interface prototyping tools is presented in [19]. Recently, multi-device interaction where many different cross-device systems and tools have been investigated [14]. Among the aspects discussed are specific topics for the development of UIs between devices addressing problems and solutions suitable for non-technical users. In another context, we found many software

[1] Wicked nature is a term used for describe a social or cultural situation/problem that is difficult to solve because the problem is not understood until after the formulation of a solution.

development frameworks like [8], interactive applications and academic research supporting sketching activities [13,17,21]. [17] presents an extensive comparison of sketching tools considering UI design by sketching and Collaborative Design. FlexiSketch TEAM [21] allows multiple users to simultaneously use their own tables to quickly gather and validate requirements, for which UCD can be used. [4] concludes that paper prototypes have been found to be useful in practice in many more ways than just for usability testing. [13] puts forward the relationship between the agile practices and UI design. Finally, [7] shows the importance of using UCD as an early practice in the SDLC. The inclusion of the collaborative method within UCD and Agile is aligned with previous proposals, such as Communication-Centered design and Extreme Designing [18], where the focus is on the communication between designers and users.

3 A Collaborative User Centered Design Method Based on Sketching

The CUCD method is derived from the observation of a collection of user-centric methods concentrated on the construction of usable and ergonomic UIs (like Brainstorming, Paper Prototyping, etc.). Theses activities engages the user, its activities and their environment in all stages of an interactive application's analysis and design. It involves the activities of Drawing, Prototyping, Sharing/Testing and Discussing/Reflecting. In the **Drawing**, designers define the structure of the interface, producing some sketches depicting a UI in the form of screens, roughly one for each *"state"* of the UI. This activity is then iteratively performed between the elicitation of requirements and design activities when designers and users need to obtain the system specifications. Designers and end-users would sometimes be divided into subgroups. An aspect very important to achieve as it facilitates the collaborative sessions between stakeholders and end-users. **Prototyping** allows designers to define the **behavior** by assembling the different screens in a logical manner. In the **Sharing/Testing** normally, each subgroup sticks their produced prototype on a wall, arranged in a logical way, forming a sort of storyboard. An overall validation happens at that stage, where the navigation flows are tested. Very often, the end-users point to the drawings following the previously designed navigation, in a sort of *"Storytelling"*. The last stage is the **Discussing/Reflecting** where once the design is discussed and the possible problems are highlighted, the modifications are agreed. Then, designers and end-users proceeds to another iteration of drawing/prototyping/sharing until both parts are satisfied with the results.

The observations of UCD activities served as the fundamental concepts for devising the method and consequently the requirements for the tool:

1. *A software tool for design sessions.* After the design sessions, the designers would make a report that contains details about what was produced and decided. This is the first point in which a software tool could be of use, since it would register automatically everything that is produced on a session.

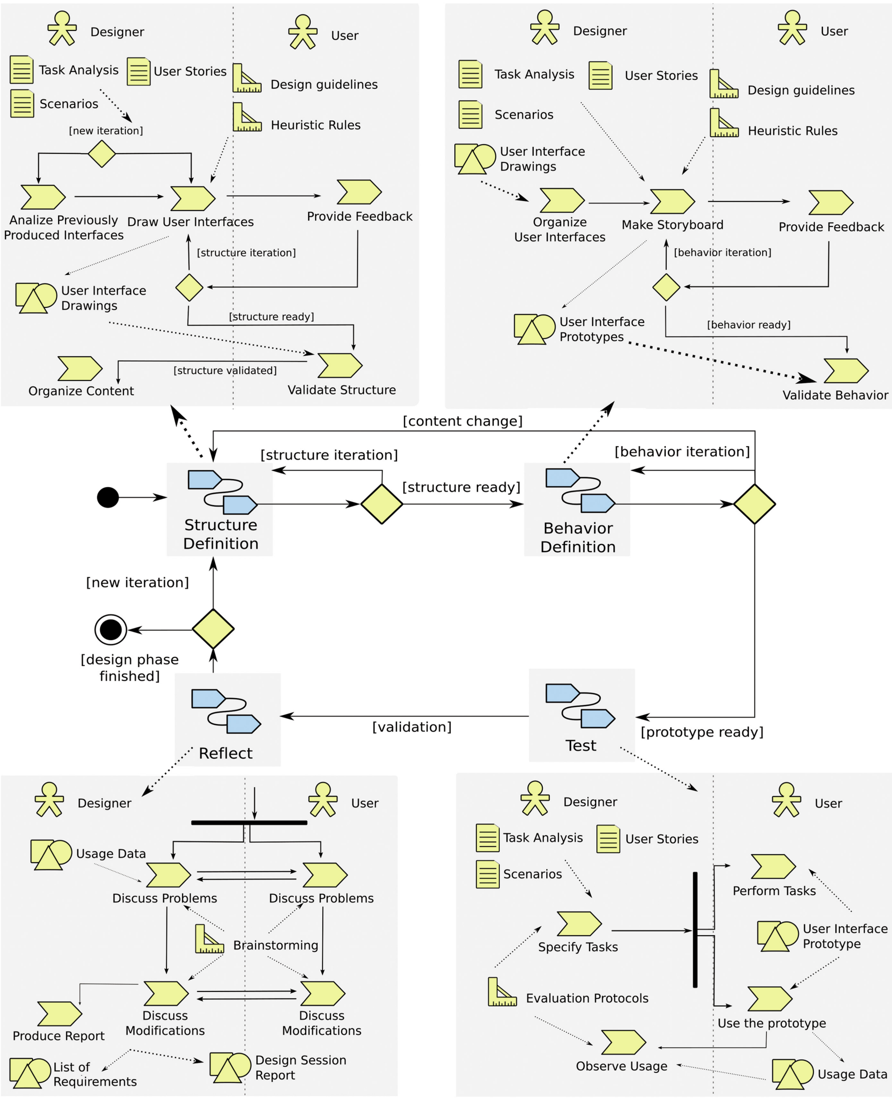

Fig. 2. The method workflow.

2. *Aparté sketching.* Often, some participants may feel uncomfortable about sketching in front of the rest of the group. For those participants, the designer usually gives a private sheet of paper, to be placed on the wall after the user feels his/her design view is good enough for discussion. This Aparté (aside) sketching activity would engage users as an active part of the solution design, and not only for validation.

3. *Different devices and fidelities.* Despite the fact the designers most part of time uses only pen and paper to conduct their design's sessions, they would frequently bring different media according to the users' needs. For instance, they would bring "pixel-perfect" (high-fidelity) software tools like Axure and Visio if they needed to design the final look of a system. The users also usually provided picture of interfaces previously done, in case of a redesign and often bring in their own devices or devices that are important for simulating the usage of the system in a real situation.
4. *Realistic prototypes.* Sometimes it's hard to imagine a design for a specific platform without seeing what it would look like on it. A lot of redesign could be avoided simply by drawing the interface on the very device it is indented to run. In other words, the physical domain would enable designers to make use of the device's properties that are often lost or ignored using just a paper simulation.

Based on the UCD activities and the fundamental concepts, the CUCD method is divided into four basic phases: *Structure Definition*, *Behavior Definition*, *Test Definition*, and *Reflect Definition*. The goal of devising such a method is to support the activities identified on the paper prototyping practice, keeping in mind the basics of Sketching and Prototyping as tools for helping designers and users to communicate, and also to include technological support. The workflow that links all the disciplines together is presented in Fig. 2.

The **Structure Definition Phase** allows Designers to define the structure of the interface, depicting a UI from screens either by sketching them or by composing interface elements according to the gathering requirements. All the elements of the UI are set according to the interaction needs of the user and to the functional specifications. In this step, which can be conducted while interviewing the user, the Designer concentrates on the content of the interface and on what is going to be shown. The activities begin with the definition of the structure by designers and users and iterate **([structure iteration]/[structure ready])** proceeding to the next step when both parts agree. Figure 2 details this step further, having the Designer starting the process by either Analyzing Previously Produced Interfaces and Drawing UI or simply by drawing them. For those activities, the Designer uses models such as User Stories, Task Analysis, Scenarios and Guidelines such as Design Guidelines and Heuristic Rules. The User should provide feedback about the produced interfaces until the structure is agreed and validated. Then, the Designer should organize the content in order to proceed to the next step.

In the **Behavior Definition Phase** Designers are focused on the practical usage of the system, or how the information will be shown, while the different screens are assembled in a sequential way (Fig. 2). The activities start with the Designer assembling the different UIs produced previously and then proceed to Make the Story-board which connects the interfaces in a logical manner, taking into consideration the models produced in the Structure Definition. The UI Prototypes are created in this activity for further validation by the User.

In the **Test Definition Phase** each subgroup puts their prototypes on the wall, arranged in a logical way, forming a sort of storyboard (Fig. 2). This step requires the active participation of the User, since this is the role that validates the prototype. The Designer specifies the task to be conducted, that can match the Scenarios, User Stories, Task Analysis, etc. to be evaluated, either by planning a user study with Evaluation Protocols. The User should perform the tasks while using the prototype, and the Designer has the role of observing the usage, from which usage data might be collected and evaluated.

In the **Reflect Definition Phase** Designers and Users can use brainstorming techniques to discuss possible problems and modifications. This activity starts by splitting the work between designers and users in parallel. Nevertheless both perform essentially the same activity, since they discuss problems and modifications. Designers have the responsibility to produce a report containing a Requirement List (regardless of whether the overall iteration is over or not). The list can be seen as a series of post-it notes stuck to the wall. In case the iteration is over, a design session report is produced, otherwise a new iteration can begin.

3.1 Discussion

Flexible software development encompasses a series of methods for software development that, in overall, consider software design as a problem of *wicked nature*. One of the highest priorities in flexible and ASD process is the development of runnable software that can be validated by both Stakeholders and Users. Generally, a prototype is used as a proof of concept. It can be considered as a first model towards a software development process. The holistic view in ASD is an iterative process of discovery, coding and validation, with small iterations. This vision does not consider that system requirements will all be provided correctly. Our method is a recommendation for rapid gathering and validating requirements, for which UCD can be used.

The CUCD method is presented independently of any technological support. However, offering a tool support for the method considering the collaboration among distributed teams requires also to take into account a multi-surface context where designers and final users they could working together using multiple devices to interact. In order to show how to include multiple devices or surfaces on the method, a combination of the configurations can be defined which considers the different users (or roles, which contains both designers and users) and devices. We can consider single and multiple users and also, single and multiple devices, thus giving a combination of 4 possible configurations of roles and devices (see Fig. 1). Each step of the method can use one of the 4 possible *"user X device"* combinations, being expressed by n^r, where n is the number of combinations and r is the steps of the method. In total, there are 256 possible combinations of single/multiple users, single/multiple devices and the phases Structure definition, Behavior definition, Test, Reflect.

4 The Collaborative Sketching System

The Collaborative Sketching System (CSS) is "a multi-surface collaborative tool for UI design that allows the sketching and simulation of UIs on many different devices". The tool was developed using a web technology - Google Web Toolkit (GWT). Therefore, it is accessible online and works on almost every modern web browser which makes it reachable from various platforms and interaction surfaces. The CSS offers a distributed multi-surface workspace. It provides users to interact around a UID method. Users can participate by using their personal device or different types of devices. These devices ranging from tiny to large interactive surfaces at the same time. Figure 3 illustrates a use of the CSS on the same session with a phone, a tablet and a tabletop computer. The main capabilities and features of the tool are: (1) Multi-surface and collaboration; (2) Infinite workspace and (3) Supports the four step of the CUCD method.

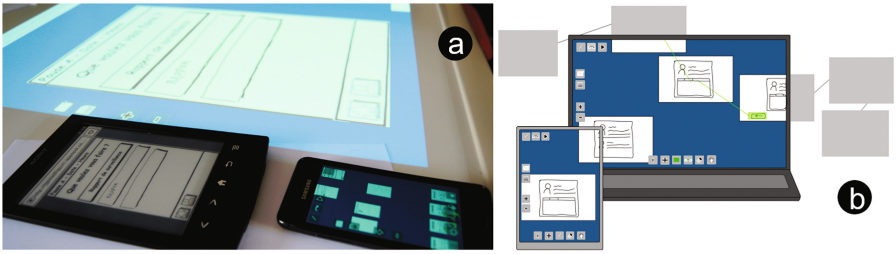

Fig. 3. (a) A collaborative sketching session on a phone, a tablet and a tabletop computer. (b) The infinite workspace concept of the collaborative sketching system.

The CSS allows the user to draw on virtual sheets of paper on a wall, that is represented as an infinite workspace (Fig. 3). Many devices can be used at the same time as they show parts of that workspace in an analogous way to a map software. Users can zoom in/out to show more or less details, sketch interfaces and "play" them as if they were real systems.

In the CSS the "Interface production" (Fig. 4) is conducted mainly by designers. It eventually includes users for providing feedback, validating the structure and optionally producing themselves the interface drawings. By activating, the Sketch mode, it is possible for the user to create "scenes" and sketch on them. The designer can add, remove and organize scenes spatially. The CSS includes an import feature that allows the possibility for the user to construct interfaces in different levels of fidelity: low (sketching), medium (wireframe mockup) and high (UI screenshot).

The "Prototype production or Behavior definition" (Fig. 4) is also conducted mainly by designers and eventually includes users for providing feedback, validating the behavior and optionally producing prototypes themselves. With the Prototype mode, the user is able to create interactive zones on top of the scenes (e.g. a "next" button) and connect these zones to another scene existing in the

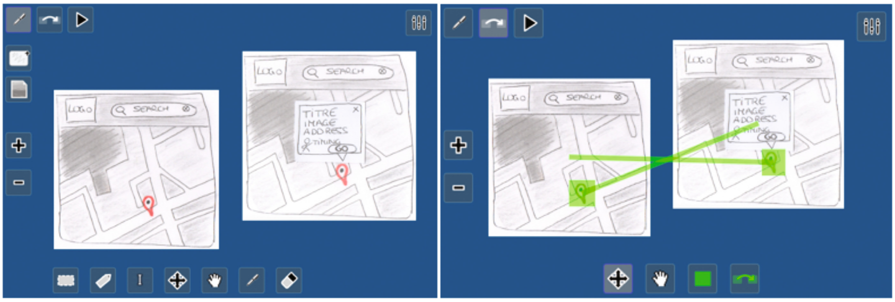

Fig. 4. The interface production (left) and the behavior definition (right) of the collaborative sketching system

defined structure (e.g. the scene that is targeted by the "next" button). The last feature consists in the test of the developed prototype in any device. By selecting the "Play" mode, designers can select the UI where he wants to start and the tool will show only the selected scene in full screen. The prototype might be navigated since the defined interactive zones will react to users input, showing the next scene when an action occurs.

4.1 Experiment

We have conducted a usability study with 20 participants having different backgrounds, (11 female or 55%), aged between 15 and 57 years old ($M = 34.25$, $SD = 10.26$). Participants were divided into 2 groups and instructed to sketch a system UI for children. An IBM Computer Satisfaction Usability (IBM CSUQ) Questionnaire [11] was completed by the participant to give their feedback about the CSS. Five additional questions were added to IBM CSUQ concerning the usage of animation pictures and UI navigation concerns: (20) "For me, animated images are not recommended"; (21) "I always know where I am and how to go where I want to"; (22) "Returns and outputs are explicit and visible"; (23) "The address of the current page allows me to return easily"; (24) "The caption or alternative text allows me to understand the meaning of the image"; (25) "The colors are chosen in order to leave the information readable".

Figure 5 depicts acceptable results of a user trial applied to 20 participants. Both results of the SysUse show that the groups in general have been appreciated. The results suggest that the CSS provides the support required for our CUCD method based on sketching. The InfoQual also suggests that participants from both groups have appreciated the quality of the information manage by the CSS. An aspect that we will to consider is the possibility to incorporate a history of annotations into the scenes useful when final users perform the simulations of the interfaces. The InterQual shows that the user interface of CSS is acceptable for the participants. However, we consider that the tool requires additional improvements for example: the possibility to manage versions of the same document and the possibility to testing the behaviour of the user interfaces without having to access the interface production. The Overall satisfaction suggests that CSS is positively perceived for the participants of the groups. The system overall usability suggests that the participants from both groups agree with the statements made in the survey. Table 1 shows cumulated results of the IBM CSUQ for the most significant questions questions according to the responses. Group 1 responded that the information provided in the CSS is easy to understand (Q13), while group 2 responded that the CSS is simple to use (Q2). Regarding the least voted questions Group 1 disagrees that the moving images are not recommended (Q20). Finally, Group 2 stated that they did not believe that they were productive quickly using CSS (Q8).

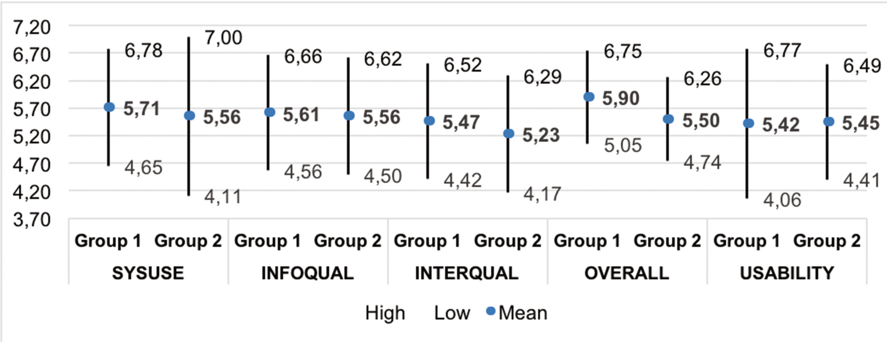

Fig. 5. The IBM CSUQ results.

Table 1. Cumulated results of the CSUQ. (1) The most suffrage question; (2) The less suffrage question; (3) The most critical participant; (4) The less critical participant

Question	Group 1	Group 2
(1)	**13** ($\mu = 6,15$; $M = 6,00$; $\sigma^2 = 0,51$; $\sigma = 0,67$)	**2** ($\mu = 7,00$; $M = 7,00$; $\sigma^2 = 0,00$; $\sigma = 0,00$)
(2)	**20** ($\mu = 4,30$; $M = 4,00$; $\sigma^2 = 1,66$; $\sigma = 2,05$)	**8** ($\mu = 3,25$; $M = 3,00$; $\sigma^2 = 0,73$; $\sigma = 0,91$)
(3)	**6** ($\mu = 4,64$; $M = 4,00$; $\sigma^2 = 0,91$; $\sigma = 1,04$)	**20** ($\mu = 4,40$; $M = 5,00$; $\sigma^2 = 1,47$; $\sigma = 1,85$)
(4)	**2** ($\mu = 6,96$; $M = 7,00$; $\sigma^2 = 0,08$; $\sigma = 0,20$)	**2** ($\mu = 6,04$; $M = 6,00$; $\sigma^2 = 0,84$; $\sigma = 1,02$)

5 Conclusions and Future Work

We have presented a CUCD method based on sketching to support early prototyping along flexible and ASD process. It is presented as independent of technological support, which is then added in order to support the assessed requirements. The approach is far from introducing new software development methodology that integrates UI engineering with the process, but is rather an attempt to include the steps of the proposed method into existing practices through UCD. The CSS, constructed to support the method, takes into consideration the contemporary technological support of multiple-surface. Its tools have the original property of functioning with multiple possible combinations of designers, users and devices at the same time. Future works will focus on describing how the UIs designed by sketches with our CUCD method can be linked with others activities that happen within a regular software development process. In this view, we consider performing several experimental studies which involve classical known SE methodologies to find new insights and improvements to our method.

References

1. Ambler, S.W.: Agile adoption rate survey results, March 2007
2. Ambler, S.W.: Modeling and documentation practices on it projects survey results, July 2008
3. Ambler, S.W.: 2013 agile project initiation survey results (2013)
4. Aston, J., Meszaros, G.: Adding usability testing to an agile project. In: Proceedings of Agile 2006 Conference (AGILE 2006) (2006)
5. Bodart, F., Hennebert, A.M., Leheureux, J.M., Provot, I., Vanderdonckt, J., Zucchinetti, G.: Key activities for a development methodology of interactive applications, pp. 109–134. Springer, London (1996)
6. Buxton, B.: Sketching User Experiences: Getting the Design Right and the Right Design. Morgan Kaufmann Publishers Inc., San Francisco (2007)
7. Fox, D., Sillito, J., Maurer, F.: Agile methods and user-centered design: how these two methodologies are being successfully integrated in industry. In: Agile 2008 Conference, pp. 63–72. IEEE (2008)
8. Gonzalez-Perez, C.: Filling the voids-from requirements to deployment with OPEN/Metis. In: ICSOFT (1), p. 19 (2010)
9. Landay, J.A., Myers, B.A.: Interactive sketching for the early stages of user interface design. In: Proceedings of the CHI 1995, pp. 43–50. ACM Press (1995)
10. Landay, J.A., Myers, B.A.: Sketching interfaces: toward more human interface design. Computer **34**(3), 56–64 (2001)
11. Lewis, J.R.: IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. Int. J. HCI **7**(1), 57–78 (1995)
12. Van der Lugt, R.: Functions of sketching in design idea generation meetings. In: 4th Conference on Creativity & Cognition, pp. 72–79. ACM (2002)
13. McInerney, P., Maurer, F.: UCD in agile projects: dream team or odd couple? Interactions **12**(6), 19–23 (2005)
14. Nebeling, M., Kubitza, T., Paternò, F., Dong, T., Li, Y., Nichols, J.: End-user development of cross-device user interfaces. In: 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 299–300. ACM (2016)
15. Olson, G.M., Olson, J.S.: Distance matters. HCI **15**(2), 139–178 (2000)
16. Ossher, H., van der Hoek, A., Storey, M.A., Grundy, J., Bellamy, R.: Flexible modeling tools (FlexiTools 2010). In: 32nd ACM/IEEE International Conference on Software Engineering-Volume 2, pp. 441–442. ACM (2010)
17. Sangiorgi, U.B., Beuvens, F., Vanderdonckt, J.: User interface design by collaborative sketching. In: Proceedings of the Designing Interactive Systems Conference, pp. 378–387. ACM (2012)
18. da Silva, B.S., Aureliano, V.C.O., Barbosa, S.D.J.: Extreme designing: binding sketching to an interaction model in a streamlined HCI design approach. In: 7th Brazilian Symposium on Human Factors in CS, pp. 101–109. ACM (2006)

19. Silva, T.R., Hak, J.L., Winckler, M.: A review of milestones in the history of GUI prototyping tools. In: 15th IFIP TC. In: 13th International Conference on Human-Computer Interaction, vol. 22, p. 267. University of Bamberg Press (2015)
20. Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., Heldal, R.: Industrial adoption of model-driven engineering: are the tools really the problem? In: International Conference on MDE Languages and Systems, pp. 1–17. Springer (2013)
21. Wuest, D., Seyff, N., Glinz, M.: Sketching and notation creation with FlexiSketch team: evaluating a new means for collaborative requirements elicitation. In: 23rd International Requirements Engineering Conference, pp. 186–195. IEEE (2015)