A Tool for Multi-Surface Collaborative Sketching

Jorge Luis Pérez-Medina
Université catholique de Louvain
Louvain School of Management
Research Institute
Place des Doyens, 1-B-1348
Louvain-la-Neuve (Belgium)
jorge.perezmedina@uclouvain.be

Jean Vanderdonckt
Université catholique de Louvain
Place des Doyens, 1-B-1348
Louvain-la-Neuve (Belgium)
jean.vanderdockt@uclouvain.be

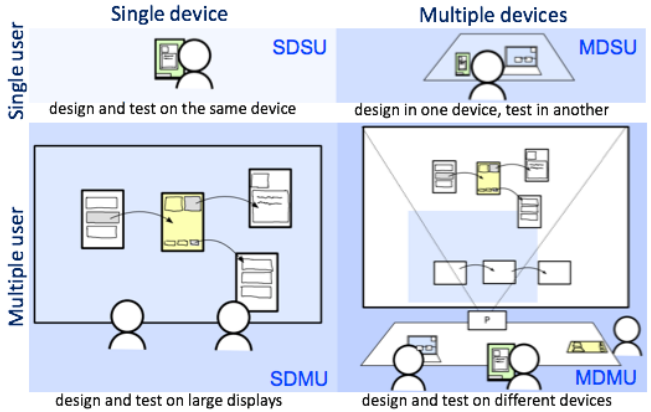

Figure 1: Four physical configurations of a Collaborative User Centered Design Method considering multiple users and devices: SDSU=single device and single user; SDMU = single device and multiple users; MDSU= multiple devices and single user; MDSD= multiple devices and multiple users.

Presented at the Cross-Surface'16 workshop, in conjunction with ACM ISS'16.
November 6, Niagara Falls, Canada.

Abstract
This research paper focus on Collaborative Sketching as a tool to facilitate the User's Interface prototyping with multi-surface required by stakeholders and final users in early stages of the software development when developers have a vague idea in mind of what should the service or system being developed or improved will be. The paper also offers a Collaborative User Centered Design method based on sketching.

Author Keywords
Electronic sketching; Multi-surfaces; Collaborative design; Prototyping.

ACM Classification Keywords
H.5 [INFORMATION INTERFACES AND PRESENTATION (e.g., HCI)]: H.5.2 User Interfaces - Graphical user interfaces (GUI) - Prototyping - User-centered design

Introduction
Software modeling tools are often not used during early steps of the software life-cycle for which they would be more helpful [11]; free-form approaches, such as office tools, white boards and papers arranged on a wall, are frequently used instead [14]. We agree with the recommendations expressed by [14]. They suggest the SE community to pay more attention to use the tried-and-tested methods.

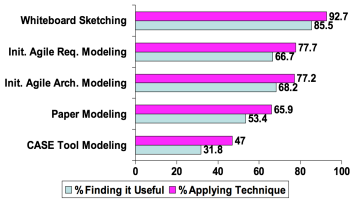

Figure 2: Primary approaches to modeling [1].

A survey realized by [1] summarize the way how effective initial requirements modeling was on agile teams. Figure 2 shows that 92.7% of respondents indicated that those teams were using *Witheboard Sketching* for requirements gathering, and that 85.5% of those teams found the effort worthwhile.

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's Interface design throughout the software life-cycle.

The User Interface is the portion of software with which a user directly interacts. Designers can explore the requirements using essential UI prototypes [1]. It represent UI requirements in a technology independent manner, just as essential use case models do for behavioral requirements. Although essential UI prototyping is a great technique, 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. For example, whiteboard diagrams typically suffice for initial requirements models.

We aim at providing support for collaborative and flexible modeling in early steps of a User Centered Design process where stakeholders and final users can collaborate by using multi-surface provided by multi-device. It consists of using sketching in early UI design as an informal and flexible manner for producing any model for requirements and analysis phases. 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 UI design tools, used in HCI, generally support electronic sketching, giving more freedom to change sketches and more flexibility in creating and evaluating a design prototype [8]. The UI development process, is usually conducted in HCI. It appears to be eminently informal to support collaborative activities involving stakeholders and final users.

By using sketches designers may be more motivated, more creative, and perhaps better able to address the challenges of creating a more successful design, and thus produce a better design outcome [16]. 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 operation of a tool can be described to a non-computer user with greater ease. Prototyping is characterized by collaborative interactions that inherently influence how design is actually practiced, therefore a tool to support User Interface Design should ideally be flexible in order to accommodate those design situations, i.e, different activities, people from different areas and using different media and devices. We propose a combination of 4 possible configurations presented in Figure 1 which considers the different users (or roles which contains both designers and users) and devices.

This paper presents a flexible Collaborative User Centered Design (CUCD) method based on sketching to support prototyping along a Software Development process. It promotes flexible communication among stakeholders and end-users in early steps of development when it is vital to understand the system requirements. The remainder of this paper is structured as follows. In section 2, sketching and basic concepts of design by sketching are presented. Section 3 presents some related work. Section 4 describes a (CUCD) method considering that all its actors have potential to participate in the prototyping process. Section 5 introduces GAMBIT, a tool supporting the method. Section 6 presents the results of a pilot experiment. Finally, conclusions and some remarks for future work are presented.

Design by Sketching

Sketches are rapid drawings [3]. Sketching is useful as a design tool. Stakeholders and final users are able to carry out UI design by sketching during collaborative sessions of brainstorming of ideas to understand the requirements.

Design is a hard challenging for designers. 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, i.e. the process of solving it is identical with the process of understanding it.

Scott Ambler in 2003 defines UI Prototyping as an iterative analytical technique in which users are actively involved [1]. With UI prototyping, the final user is able to provide valuable feedback from early stages of development.

In wicked problems, the designer does not have a clear understanding of what to produce. During the initial phases of this process designers have only a vague goal in mind. Sketching stimulates a re-interpretive cycle in the individual designer's idea generation process [13]. 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.

In the HCI domain, UI design 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 UI design is included in more general methods such as User Centered Design which has specific roles that attribute different responsibilities to the party involved. The activity of UI sketching has several recognized virtues, mainly for maintaining a flexible representation of a design, which is reckoned to foster creativity [2].

Related Work

In literature review, we found many software development frameworks like [6], interactive applications and academic research supporting sketching activities. Efforts address mainly tools and methods. [12] presents a extensive comparison of sketching tools considering UI design by sketching and Collaborative Design.

FlexiSketch TEAM [15] is a solution for collaborative, model-based sketching that allows multiple users using their own tables to work on the same model sketch simultaneously and use lightweight meta-modeling mechanics to collaboratively define custom notations on the fly. It considers a mechanism for quickly gathering and validating requirements, for which User Centered Design (UCD) can be used.

[10] concludes that paper prototypes have been found to be useful in practice in many more ways than just for usability testing. [9] presents a vision of how UCD and agile practices coexist in a development team. It puts forward the relationship between the agile practices and UI design, for example: on agile practices there is little notion of a "UI designer" specialty, but, on HCI domain, UI designers practices may be more of a *"team effort"*. Agile projects could require UI design, which typically iterates only on the UI using low technology prototypes. Finally, [5] shows the importance of using UCD as an early practice in the software development lifecycle. The inclusion of the collaborative method within UCD and Agile is aligned with previous proposals, such as Communication-Centered design and Extreme Designing [4], where the focus is on the communication between designers and users. Those approaches advocate that instead of having user stories and sketches as the only design documentation, the use of a streamlined structuring representation that binds together the stories and sketches would benefit the process of designing interfaces.

A Collaborative User Centered Design method based on sketching

This section presents a (CUCD) method for prototyping based on sketching, enabling fast, flexible, intuitive and reusable prototype. The method is derived from the observation of UCD practices (like Brainstorming, Paper Prototyping, etc.) used by two Belgian companies. Both companies observed are using Paper Prototyping [7], which enables design UIs. Figure 3 on page 4 represents a Paper Prototyping session with a cycle of four different activities.

User Centered Design (UCD) represents a collection of user-centric methods and more generally a philosophy for approaching technology design. UCD methods are not isolated activities of development process. Theses activities are concentrated on the construction of usable and ergonomic UIs. UCD engages the user, its activities and their environment in all stages of an interactive application's analysis and design.

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 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 interface. 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. The second phase covers **Prototyping** where designers defined the **behavior** by assembling the different screens in a logical manner either in a flip-book or in a sequence of screens linked with arrows in order to make the navigation explicit.

The next step is the activities of **Sharing/Testing**. At this stage, 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 participants 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 the group proceeds to another iteration of drawing/prototyping/sharing until both parts are satisfied with the results.

Based on these observations, our collaborative method is divided into four basic phases: *Structure Definition*, *Behavior Definition*, *Test Definition*, and *Reflect Definition*. The goal of devising such 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. From the Drawing activity, we have

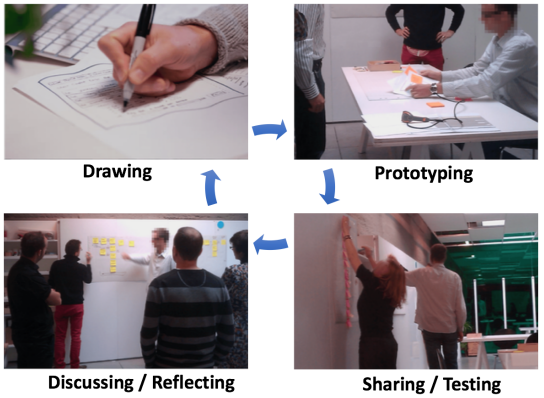

Figure 3: UI design sessions based on Paper Prototyping.

abstracted the Structure Definition, from the Prototyping we have abstracted Behavior Definition, and respectively from Sharing/testing to Test and Discussing/Reflecting to Reflect. The work-flow that links all the disciplines together is presented in Figure 4.

The Structure Definition Phase

Designers define the structure of the interface, depicting a UI from screens either by sketching them or by composing interface elements. Sketches are created according to the gathering requirements. All the elements of the UI are set according to the interaction need 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 4 details this step further, having the

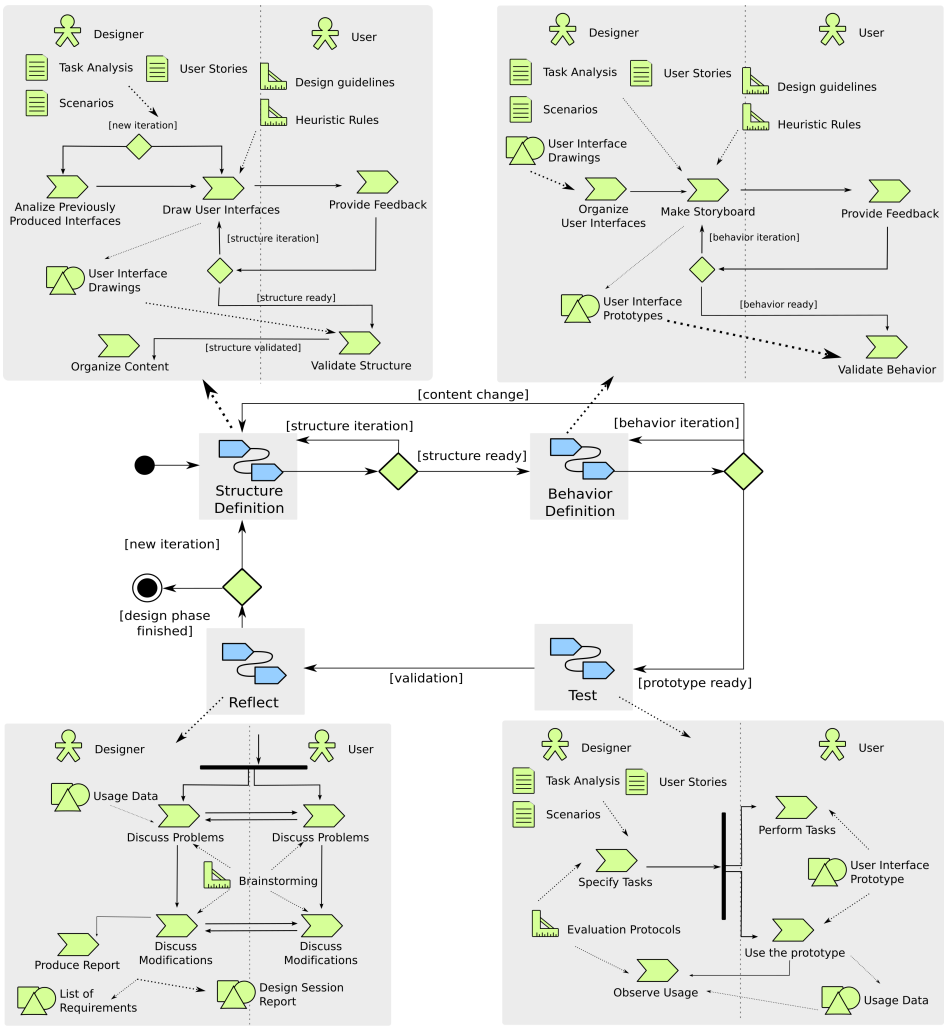

Figure 4: The Method Workflow.

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.

The Behavior Definition Phase

In this phase, Designers focuse on the practical usage of the system, or how the information will be shown, while the different screens are assembled in a sequential way (see Figure 4). 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. They might also iterate back to Structure definition prior to proceeding to the next step, since the definition of behavior very often leads to insights about the content that will be displayed.

The Test Definition Phase

In this phase, each subgroup puts their prototypes on the wall, arranged in a logical way, forming a sort of storyboard (see Figure 4). 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.

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, (see Figure 3 on page 4). In case the iteration is over, a design session report is produced, otherwise a new iteration can begin.

Discussion
One of the highest priorities in flexible and agile software development 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 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. The proposed method is a recommendation for rapid gathering and validating requirements, for which UCD can be used. The inclusion of the method within UCD and Agile is aligned with previous proposals, such as *Communication-Centered* design and *Extreme Designing* [4], where the focus is on the communication between designers and users. 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 Figure 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.

Figure 5: GAMBIT session on a phone, a tablet and a tabletop computer.

GAMBIT is the acronym for Gatherings And Meetings with Beamers and Interactive Tablets. The main capabilities and features of the tool are:

- Multi-surface and collaboration.
- Infinite workspace.
- Supports the four step of the collaborative UCD.

GAMBIT

GAMBIT is "a multi-surface collaborative tool for UI design that allows the sketching and simulation of UIs on many different devices" [12]. 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. Basically if the device has a sufficiently updated browser application, GAMBIT will run on it. Figure 5 illustrates this point showing a use of gambit on the same session with a phone, a tablet and a tabletop computer.

GAMBIT allows the user to draw on virtual sheets of paper on a wall, that is represented as an infinite workspace (Figure 6). 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 GAMBIT the "Interface production" (see Figure 7) 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 Skectch mode, it is possible for the user to create "scenes" and sketch on them. The designer can add, remove and organize scenes spatially. GAMBIT includes an import feature allows lets the possibility for the user to con-

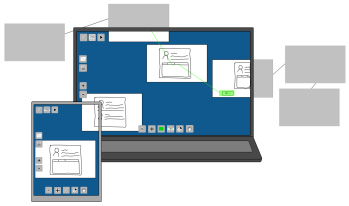

Figure 6: The infinite workspace concept of the GAMBIT

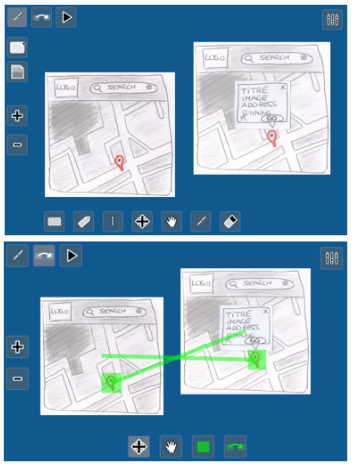

Figure 7: The Interface production (above) and the Behavior Definition (down) of GAMBIT.

struct interfaces in different levels of fidelity: low (sketching), medium (wireframe mockup) and high (UI screenshot).

The "Prototype production" (see Figure 7) 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 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. The designer can select the screen where he wants to start and the tool will remove all other interface controls showing only the selected scene in full screen. The prototype can be navigated since the defined interactive zones will react to users' input, showing the next scene when an action occurs.

Experimentation

Table 1 shows acceptable results of a user trial applied to 20 participants having different backgrounds. Participants were divided into 2 groups and instructed to sketch a system UI for children. Participants are asked to fill the IBM Computer Usability Questionnaire (CSUQ) to give their feedback about GAMBIT. The IBM CSUQ was increased with questions concerning the usage of animation pictures, and UI navigation concerns. Group 1 responded that the information provided in GAMBIT is easy to understand (Q13), while group 2 responded that GAMBIT is simple to use (Q2). Regarding the least voted questions Group 1 disagrees that the moving images are not recommended (Q20). Finally, the Group 2 stated that they did not believe that they were productive quickly using GAMBIT (Q8).

	Group 1	Group 2
(1)	13 ($\mu = 6,15; \sigma = 0,67$)	2 ($\mu = 7,00; \sigma = 0,00$)
(2)	20 ($\mu = 4,30; \sigma = 2,05$)	8 ($\mu = 3,25; \sigma = 0,91$)
(3)	6 ($\mu = 4,64; \sigma = 1,04$)	20 ($\mu = 4,40; \sigma = 1,85$)
(4)	2 ($\mu = 6,96; \sigma = 0,20$)	2 ($\mu = 6,04; \sigma = 1,02$)

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.

Conclusion and future works

We have presented a CUCD method based on sketching to support early prototyping along flexible and agile software development process. 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 User Centered Design. The method is presented as independent of technological support, which is then added in order to support the assessed requirements. GAMBIT, 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.

Acknowledgement

The authors would like to thank Ugo Sangiorgi for the big contribution to the GAMBIT tool.

REFERENCES

1. AMBLER, S. W. Agile adoption rate survey results: March 2007, 2007.
2. BELLAMY, R., DESMOND, M., MARTINO, J., MATCHEN, P., OSSHER, H., RICHARDS, J., AND SWART, C. Sketching tools for ideation (nier track). In *Proc. of the 33rd Int. Conf. on Software Engineering* (New York, NY, USA, 2011), ICSE '11, ACM, pp. 808–811.
3. BUXTON, B. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
4. DA SILVA, B. S., AURELIANO, V. C. O., AND BARBOSA, S. D. J. Extreme designing: Binding sketching to an interaction model in a streamlined hci design approach. In *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems* (New York, NY, USA, 2006), IHC '06, ACM, pp. 101–109.
5. FOX, D., SILLITO, J., AND MAURER, F. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In *Agile, 2008. AGILE '08. Conference* (Aug 2008), pp. 63–72.
6. GONZALEZ-PEREZ, C. Filling the voids - from requirements to deployment with open/metis. In *ICSOFT 2010 - Proceedings of the Fifth Int. Conf. on Software and Data Technologies, Volume 1, Athens, Greece, July 22-24, 2010* (2010), p. 19.
7. LANCASTER, A. Paper prototyping: The fast and easy way to design and refine user interfaces. *Professional Communication, IEEE Transactions on 47*, 4 (Dec 2004), 335–336.
8. LANDAY, J. A., AND MYERS, B. A. Sketching interfaces: Toward more human interface design. *Computer 34*, 3 (2001), 56–64.
9. MCINERNEY, P., AND MAURER, F. Ucd in agile projects: Dream team or odd couple? *interactions 12*, 6 (Nov. 2005), 19–23.
10. MESZAROS, G., AND ASTON, J. Adding usability testing to an agile project. In *AGILE* (2006), J. Chao, M. Cohn, F. Maurer, H. Sharp, and J. Shore, Eds., IEEE Computer Society, pp. 289–294.
11. OSSHER, H., VAN DER HOEK, A., STOREY, M.-A., GRUNDY, J., AND BELLAMY, R. Flexible modeling tools (flexitools2010). In *Proceedings of the 32Nd ACM/IEEE Int. Conf. on Software Engineering - Volume 2* (New York, NY, USA, 2010), ICSE '10, ACM, pp. 441–442.
12. SANGIORGI, U. B., BEUVENS, F., AND VANDERDONCKT, J. User interface design by collaborative sketching. In *Proc. of the Designing Interactive Systems Conference* (2012), ACM, pp. 378–387.
13. VAN DER LUGT, R. Functions of sketching in design idea generation meetings. In *Proc. of the 4th Conference on Creativity & Cognition* (New York, NY, USA, 2002), C&C '02, ACM, pp. 72–79.
14. WHITTLE, J., HUTCHINSON, J., ROUNCEFIELD, M., BURDEN, H., AND HELDAL, R. *Industrial adoption of model-driven engineering: are the tools really the problem?* LNCS. Springer, 2013, pp. 1–17.
15. WUEST, D., SEYFF, N., AND GLINZ, M. Sketching and notation creation with flexisketch team: Evaluating a new means for collaborative requirements elicitation. In *Requirements Engineering Conference (RE), 2015 IEEE 23rd Int.* (Aug 2015), pp. 186–195.
16. YANG, M. C. Observations on concept generation and sketching in engineering design. *Research in Engineering Design 20*, 1 (2009), 1–11.