

## Literature Review

In many of the journals, articles and books I read regarding high-fidelity vs. low-fidelity prototyping, an often-quoted study was the one conducted by Robert A. Virzi, Jeffrey L. Sokolov, and Demetrios Karis in 1996. The study was published by GTE Laboratories under the name *Usability Problem Identification Using Both Low and High-Fidelity Prototypes*.<sup>1</sup> This study concluded that low-fidelity prototyping, such as paper prototyping, was not a blunt instrument, and, in fact, was just as effective as high-fidelity prototyping at finding major problems.

However, this test was conducted 11-years ago in 1996. I wondered if its findings still held true today. Are the issues still the same? Can testers now create a high-fidelity prototype so much more quickly that the gap is closing between the two? Have other tools evolved that allow mid to high-fidelity prototyping at low costs? Has the increased complexity of modern interfaces created new challenges with which paper prototyping cannot keep up? Are the pros and cons the same today as they were 11-years ago between high and low-fidelity prototyping?

For example, in one 2003 study, I found a creative method for using a fairly high-fidelity prototype to create interactive simulations. The test designers used Photoshop layers to mimic screens the user would experience.<sup>2</sup> As a new menu was required, that layer was made visible, while the transparency of unneeded screen items was turned off. The fidelity of the prototype was determined by the design complexity chosen by the test giver. In my opinion, even low-fidelity Photoshop designs would be perceived as higher fidelity than a paper prototype by the participant. However, this approach would be limited to prototyping mostly computer-screen layouts. Traditional low-fidelity paper prototyping would still have an advantage over items that require mechanical interaction, such as a library checkout system.

I found that the Photoshop approach was sometimes used in Wizard-of-Oz methods of prototyping. According to Dow and MacIntyre,<sup>3</sup> the Wizard-of-Oz method is the "practice of using hidden operators to temporarily emulate unfinished parts of a computing system during development. In this type of simulation, the person normally referred to as *the computer* is instead referred to as *the wizard*. For example, the wizard might move the mouse icon on one Photoshop layer as the participant drags something that represents the mouse.

Dow and MacIntyre state that the Wizard-of-Oz method has been around since the early days of Human Computer Interface (HCI) prototyping; however, they believe they have taken this method into new venues and expanded it to meet the growing complexities of modern design. The term "Wizard of Oz" and "OZ Paradigm" was first introduced in 1980 by John F. ("Jeff") Kelley when he coined this terminology in his dissertation at the Johns Hopkins University [[Wikipedia](#)]. Dow and MacIntyre like to abbreviate Wizard-of-Oz as (WOz); although, I did not find other authors following this convention.

Due to the complexity of some of the systems Dow and MacIntyre were testing, they believed some of the wizard tasks were "too cognitively and perceptually challenging for one person" They concluded that they "should not only think about how Wizard-of-Oz can be utilized across time in an iterative process, but how they can be distributed across multiple wizards and multiple system modules".

The Wizard-of-Oz approach excelled in interactive media experiences. In today's web and software design, the interfaces have a cool factor of interaction that may be lost in paper prototyping, but which could be more closely captured using the Wizard-of-Oz approach, especially when multiple wizards are used to interact with the participant using sound effects, music and movement.

In another study, Steven Dow et al. (2005) state that Wizard-of-Oz prototyping "is particularly useful in exploring user interfaces for pervasive, ubiquitous, or mixed-reality systems that combine complex sensing and intelligent control logic".<sup>4</sup> The Wizard-of-Oz prototypes appear to be, in most cases, higher fidelity than traditional paper prototyping, require more wizards to implement and are very specific to the details of the stage of development for the prototype. In fact, Steven Dow et al. describe the prototypes as "throwaway tools" that are used a couple of times and do not evolve with the system. However, the benefits these prototypes provide are described by Steven Dow et al. with the same types of words simple paper prototyping uses by stating the following:

*The WOz method helps designers avoid getting locked into a particular design or working under an incorrect set of assumptions about user preferences, because it lets them explore and evaluate designs before investing the considerable development time needed to build a complete prototype.*

---

Wizard-of-Oz methods have been used in prototyping simulations that require speech-understanding systems. In 2000, Rayner et al.<sup>5</sup> used Wizard-of-Oz prototyping to test computer interfaces that respond to human speech. One of the challenges they realized early was that "data obtained from human-human dialogues do not transfer straightforwardly into the design of human-machine dialogues; even with perfect tuning, present-day systems cannot hope to mimic a human conversational partner."

In my experience, speech-recognition has advanced greatly in the last 8 years since this study. Both [Microsoft](#) and [AT&T Labs](#) can provide over-the-counter plug-in solutions that may be layered into a Wizard-of-Oz test. While the conversational tone has yet to be mastered, the participant can now be given a broader range of phrases and key words to use during a test. However, it appears the limitations of using human wizards to fill in for computer interactions still exist. It requires very careful controls on both the part of the participant and the wizards to collect accurate data. Participants will naturally deviate from set phrases and wordings and wizards will naturally respond, if even to commiserate with the participant.

Rayner et al. also describe Wizard-of-Oz methods as being "laborious, time consuming and costly." Obviously, these are the things that prototyping would like to avoid. One must ask if a better approach or different type of prototyping might work better.

Regarding their own Wizard-of-Oz prototyping for computer speech recognition (Natural Language interface interaction), Dahlbäck et al. bemoaned that "a computer is not a person, and some of the differences are such that they can be expected to have a major influence on the dialogue, thus making data from human interaction an unreliable source of information for some important aspects of design, in particular the style and complexity of interaction."<sup>6</sup> They also listed challenges such as "People are flexible, computers are rigid (or consistent); people are slow at typewriting, computer output is fast; computers never make small mistakes (e.g. occasional spelling errors), people make them all the time."

I wonder if paper prototyping might be more accurate at reproducing interactions requiring speech recognition? The tester could give the participant a stack of cards with acceptable words and phrases that the system reliably recognized, and the participant could use the cards to solve tasks. This would take many of the human-factor variables out of the test and alleviate the errors that plague the Wizard-of-Oz approach for speech recognition prototyping.

Because of my experience with it, I am keenly interested in the use of Flash programming to create prototyping. Daniel Wigdor<sup>7</sup> says "Flash is a good tool for building broad prototypes, and has a scripting language that allow for a certain amount of depth. Its most common use is for quickly adding animation, sound, and video to a prototype."

In Wigdor's 47-page article, he details how to use Flash and several programming tools similar to Flash for creating a simple prototype. However, it quickly becomes apparent that the time needed to implement any of his prototypes is substantially more than the time required for paper prototyping, and Wigdor suggests no additional value for the higher-fidelity prototype.

Regarding prototyping for web design, I found a repeating theme in authors who use Flash for high-fidelity prototyping. Basically, Flash is often not cost-effective for prototyping, especially if paper prototyping can be effectively substituted. Proctor et al.<sup>8</sup> wrote the following:

*High-fidelity prototypes try to mimic the look and feel, as well as the functionalities, of a real system. They are typically developed using a software tool such as Macromedia Director or Microsoft Visual Basic...In a Web environment, it has been argued that it is not as cost effective to develop high-fidelity prototypes using a desktop publishing layout tool because it would probably have taken the same amount of time to implement the Web site in hypertext markup language. The ease with which a "skeleton" Web site can be quickly created using HTML makes the Web environment a "friendlier" environment for developing interactive prototypes.*

In 2006, Alex Woolley and Steve Gill, from the PAIPR research group, National Centre for Product Design and Development Research, conducted a high-fidelity case study at the University of Wales Institute that used Flash as one of the prototype agents.<sup>9</sup> This study excited me, and I found it very clever.

Woolley and Gill used Styrofoam to create soft models of their hand-held devices. The soft models had buttons that could be pressed to correspond to choices made. The buttons passed ASCII codes through a simple serial interface with a PC. Basically, it feed the PC the equivalent of single keyboard presses. The application on screen (either Flash based, using ActionScript or PowerPoint, using PowerPoint's Visual Basic for Applications module) was programmed to respond to specific key presses. Therefore, the participant could press a key on the hand-held prototype and see the screen change accordingly. I find this exciting because I can completely visualize how to make such a prototype. I could even set it up to so that the Flash would calculate how much time passed between button presses, allowing me to create measureable data.

Woolley and Gill provide the rational for a high-fidelity prototype over a low-fidelity prototype as follows:

*"The design of information appliances is at the crossroads of a series of design disciplines. Designers are no longer bound by the generic technology offerings of a PC and there is immense freedom available to the designer in the way product interfaces are controlled and displayed. This has caused a stronger interdependence of disciplines; the interface is dependent on the input controls specified by the product designer, and the product design is dependent on the interface devices specified by the HCI Designer and Software Engineer. It is therefore important that the design of one element is not kept in isolation from the other."*

---

I interpreted this to mean that because the input control design is dependent on the various interdependent disciplines, that a higher-fidelity prototype was needed in order to test the systems together as the same time.

Other software commonly used to create user interfaces, such a Viso, has similar strengths and weaknesses to Flash. Robin Kavanagh and John Soety wrote the following in an article about using Viso for creating prototypes:<sup>10</sup>

*Traditional paper prototyping is a useful method for testing designs, but it is rather limited. Rapid navigation between screens and pop-ups, for example, is not practical with paper prototyping. In addition, some elements, such as hotspots, are better viewed through an on-line prototype.*

---

Some of these things could be done with a paper prototype. For example, a pop-up screen could easily be reproduced with a paper method. However, hot spots and things needing to happen rapidly would benefit from a programmable interface. I have used both Flash and Viso, and, while the latter would design a mock-up interface more quickly, Flash could provide greater flexibility and provide more complex interactions.

As Robin Kavanagh and John Soety point out, in Viso you cannot drag screen items or dialog boxes. This is easy to do in Flash; however, it requires someone with ActionScript programming capabilities. This hurdle would limit many prototyping test groups attempting to use Flash as a prototyping agent.

Regarding using Viso as a prototyping environment, the authors state, "Despite the limitations, we had the best of both worlds: a cheap, quick, flexible, yet polished-looking prototype without the commitment of a programming resource!"

For me, the appearance of the prototype is of little concern in the early stages of testing. However, and as is intimated above, this bothers many people. I was surprised at how many of my classmates had difficulty getting passed this point in some of the online discussions.

It seems that test appearance matters to many, and must be taken into account as one of the environmental variables. When deciding on the fidelity level of the prototype, the tester must ascertain if the appearance is going to affect the manner in which the tester interacts with the interface. If it is a major contributing factor, a higher fidelity seems to be in order.

Julie A. Jacko et al. describe the role of cognitive architecture in deciding the fidelity of the prototype.<sup>11</sup> If the system under development is a multimodal system (requiring multiple sensory inputs), the prototypes often began with low-fidelity prototypes and then progress to high-fidelity simulation. For example, Julie A. Jacko et al. state the following on p. 293:

*When a new multimodal system is in the planning stages, design sketches and low-fidelity mock-ups may initially be used to visualize the new system and plan the sequential flow of human-computer interaction. These tentative design plans then are rapidly transitioned into a higher fidelity simulation of the multimodal system, which is available for proactive and situated data collection with the intended user population. High-fidelity simulations have been the preferred method for designing and evaluating new multimodal systems, and extensive data collection with such tools preferably is completed before a fully functional system ever is built.*

Interestingly, Jacko et al. prefer the term *precision* over *low-fidelity* or *high-fidelity* because it "refers to the content of the prototype itself, not its relationship to the final." Jacko et al. (p. 1009) speak of prototypes in three categories: *Rapid prototypes* created for a specific purpose and then thrown away (paper prototyping would fall into this category), *Interactive prototypes* used to increase the precision of the final product (Wizard-of-Oz type prototypes

fall into this category; however, paper prototypes might blend in here as well), and Evolutionary prototypes, which are iterative prototypes that become part of the final system (high-fidelity prototypes often fall into this category and are more common in software interfaces).

From reading Jacko et al., the impression is given that the greater number of prototyping methods you master and have at your disposal, the more likely you are to have the proper tool when the need arises. No one prototyping method is appropriate for all situations. However, with that said, paper prototyping seems to always have a place in early development and often continues to provide value throughout each stage of production.

In their 2007 book, Arnowitz et al.<sup>12</sup> provide useful worksheets for deciding which prototyping method to use and when it is appropriate to increase the fidelity of the prototype. The category of prototyping in which Flash would fall, they call "coded." In other words, prototyping that requires a computer programmer to be involved.

The charts provided by Arnowitz et al. sort prototypes by intended audience, the stage of development, the speed with which the prototypes need to occur, the projected longevity of the prototype's usefulness, whether the expression is conceptual or experiential ("how abstract or tangible the visual design and interaction behavior are expressed in a prototype"), whether the style is narrative or interactive, the medium through which the final product will be transmitted and the level of fidelity needed.

The following charts are provided by Arnowitz et al. (see next page):

## WORKSHEET 7.1: Select a prototyping method – Overview

Project Name:					Author:					
Project Date:					Internal Ref.:					
Product Name:					Target Release:					
Current Phase:										
Methods										
Characteristics		Card sorting	Wire frame	Story board	Paper	Blank model	Video	Wizard of Oz	Digital interactive	Coded
Audience	Internal	•	•	•	•		•		•	•
	External	•		•	•	•	•	•	•	•
Stage	Early	•	•	•		•	•			
	Midterm		•	•	•		•	•	•	
	Late								•	•
Speed	Rapid	•	•	•	•	•	•	•	•	
	Diligent						•		•	•
Longevity	Short	•	•			•				
	Medium		•	•	•		•	•	•	
	Long						•		•	•
Expression	Conceptual	•	•	•	•	•	•	•	•	
	Experiential				•		•	•	•	•
Style	Narrative		•	•		•	•			
	Interactive	•			•		•	•	•	•
Medium	Physical	•	•	•	•	•				
	Digital	•	•	•			•	•	•	•
Fidelity	Low	•	•			•				
	Medium		•	•	•		•	•	•	
	High			•			•	•	•	•

• = Supported

## WORKSHEET 7.2: Select a prototyping method – Details

Project Name:		Author:								
Project Date:		Internal Ref.:								
Product Name:		Target Release:								
Current Phase:										
Characteristics		Methods								
		Card sorting	Wire frame	Story board	Paper	Blank model	Video	Wizard of Oz	Digital	Coded
Audience	Internal	+	+	+	+	+	+	+	+	+
	External	++	+	+	++	++	++	++	+	+
Stage	Early	++	+	+	-	++	+	-	+	-
	Midterm	-	-	+	+	NA	-	+	+	-
	Late	--	--	-	-	NA	--	--	-	+
Speed	Rapid	++	+	+	+	++	-	+	+	-
	Diligent	NA	-	+	-	NA	+	+	+	+
Longevity	Short	+	+	-	-	+	--	+	--	-
	Medium	+	+	+	+	-	+	+	+	+
	Long	-	-	-	-	--	+	-	-	+
Expression	Conceptual	++	+	+	+	++	+	NA	-	NA
	Experiential	NA	-	+	++	+(process)	+	++	+	++
Style	Narrative	NA	+	++	-	+	+	-	-	NA
	Interactive	++	NA	NA	++	+	+(process)	++	++	++
Medium	Physical	+	+	+	++	++	NA	+	NA	NA
	Digital	--	+	+	NA	NA	++	+	++	++

Content		Card sorting	Wire frame	Story board	Paper	Blank model	Video	Wizard of Oz	Digital	Coded
High Fidelity	Information	++	+	--	++	-	+	+	++	++
	Interaction	NA	NA	++	++	+	+	+	-	++
	Visual Design	NA	-	--	-	-	+	+	++	++
	Editorial Content	--	-	--	++	-	+	+	+	++
	Branding	--	--	+	+	--	-	-	+	++
	System	NA	NA	NA	NA	--	-	-	==	++
Low Fidelity	Information	+	++	+	--	-	--	--	--	-
	Interaction	++	++	--	NA	--	--	--	--	NA
	Visual Design	++	++	++	+	+	-	-	+	+
	Editorial Content	++	++	++	-	-	+	+	+	NA
	Branding	++	++	++	+	+	+	+	+	--
	System	NA	++	++	++	++	+	+	NA	NA

#### Legend

++ very appropriate

+ appropriate

- acceptable

-- not practical

NA not applicable

In summary, there are many different prototyping tools available. Being aware of the differences and availability of these tools will provide testers with the best choices for the job. The conception that paper prototyping is less advanced than other methods is naive. It is like suggesting that a hammer is less advanced than a jack hammer when hanging a picture frame. It is simply one of the tools available, and there are many uses for which it is definitely the best tool in the tool box.

## Works Cited

---

<sup>1</sup>Virzi, R.A., Karis, D. and Sokolov, J.L. "Usability Problem Identification Using Both Low- and High-Fidelity Prototypes." 1996. Proceedings of Conference on Human Factors in Computing Systems: CHI '96 (Vancouver), pp. 236-243. New York: ACM Press.

<sup>2</sup>Eeaudouin-Lafon, Michel Mackay and Wendy Mackay. 2003. "52 Prototyping Tools and Techniques," The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications ed. Julie A. Jacko and Andrew Sears (Mahwah, NJ: Lawrence Erlbaum Associates, 2003), p.1018.

<sup>3</sup>Dow, Steven and Blair MacIntyre. "New Media Collaboration through Wizard-of-Oz Simulations." College of Computing Georgia Institute of Technology. 2007.

<sup>4</sup>Steven Dow et al. "Wizard of Oz Support throughout an Iterative Design Process." Vol. 4, No. 4. October–December 2005.

<sup>5</sup>Manny Rayner et al. The Spoken Language Translator. Cambridge University Press. p.132-133. 2000.

<sup>6</sup>Dahlbäck et al. "Wizard of Oz Studies - Why and How." Knowledge-Based Systems, Vol. 6, No. 4, pp. 258-266. 1998.

<sup>7</sup>Daniel Wigdor. "Building a Usability Prototype in Visual Basic, Flash, and Dreamweaver." University of Toronto, Department of Computer Science. p. 4. 2002.

<sup>8</sup>Wenli Zhu, Kim Phuong L. Vu, and Robert W. Proctor, "18 Evaluating Web Usability," Handbook of Human Factors in Web Design, ed. Robert W. Proctor and Kim-Phuong L. Vu (Mahwah, NJ: Lawrence Erlbaum Associates, 2005) p. 327.

<sup>9</sup>Woolley, Alex and Steve Gill. "Information Ergonomics Lectures for Creative Prototyping." National Centre for Product Design and Development Research, Western Avenue, Cardiff, CF5 2YB Wales. 2006.

<sup>10</sup>Kavanagh, Robin and John Soety. "Prototyping Using Visio." Usability Interface, Vol 7, No. 1, July 2000.

<sup>11</sup>Sharon Oviatt, "14 Multimodal Interfaces," The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, ed. Julie A. Jacko and Andrew Sears (Mahwah, NJ: Lawrence Erlbaum Associates, 2003) . 2003.

<sup>12</sup>Arnowitz, Jonathan and Michael Arent and Nevin Berger. Effective Prototyping for Software Makers (Interactive Technologies). 2007.