Usability Evaluation Considered Harmful (Some of the Time)

Saul Greenberg
Department of Computer Science
University of Calgary
Calgary, Alberta, T2N 1N4, Canada
saul.greenberg@ucalgary.ca

Bill Buxton
Principle Scientist
Microsoft Research
Redmond, WA, USA
bibuxton@microsoft.com

ABSTRACT
Current practice in Human Computer Interaction as encouraged by educational institutes, academic review processes, and institutions with usability groups advocate usability evaluation as a critical part of every design process. This is for good reason: usability evaluation has a significant role to play when conditions warrant it. Yet evaluation can be ineffective and even harmful if naively done 'by rule' rather than 'by thought'. If done during design brainstorming, it can kill creative ideas that do not conform to current interface norms. If done prematurely during early system design, the many interface issues seen can kill what would could have been an inspired vision. If done to verify an academic prototype, it may incorrectly suggest a design's worthiness rather than offer a meaningful critique of how it would be adopted and used in everyday practice. If done without regard to how cultures adopt technology over time, then today's reluctant reactions by users will forestall tomorrow's eager acceptance.

Traditional usability evaluation should not be used to validate very early design stages or culturally-sensitive systems. Other reflective and critical methods should be considered in their stead.

Author Keywords
Usability testing, interface critiques, sketching, cultural artifacts, teaching usability.

ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces (Evaluation/Methodology).

INTRODUCTION
Usability evaluation is one of the major cornerstones of user interface design. This is for good reason. As Dix et al., remind us, evaluation helps us "assess our designs and test our systems to ensure that they actually behave as we expect and meet the requirements of the user" [7]. This is usually accomplished through several means, each with varying purposes.

Within product groups, for example, the team can evaluate products under development for 'usability bugs', where developers are expected to correct the significant problems found. Evaluation can also form part of an acceptance test, where human performance is measured quantitatively to see if it falls within an acceptable criteria (e.g., time to complete a task, error rate, relative satisfaction). Or if the team is considering purchasing one of two competing products, evaluation can determine which is better at certain things. Alternately, evaluation of existing work practices can help the team identify tasks and contextual constraints that form part of a software engineering requirements analysis.

Within HCI research and academia, evaluation validates novel design ideas and systems, usually by showing that human performance or work practices are somehow improved when compared to some baseline set of metrics (e.g., other competing ideas), or that people can achieve a stated goal when using this system (e.g., performance measures, task completions), or that their processes and outcomes improve.

Clearly, user evaluation is helpful for many situations. Indeed, we (the authors) have advocated and practiced usability evaluation in both research and academia for many decades. We believe that practitioners should continue to evaluate for most – *but not all* – interface development situations. What we will argue is that there are *some* situations where evaluation can be considered harmful: practitioners have to recognize these situations, and they should consider alternative methods instead of blindly following the evaluation doctrine. Evaluation, if wrongfully applied, can quash potentially valuable ideas early in the design process, incorrectly promote poor ideas, misdirect developers into solving minor vs. major problems, or ignore (or incorrectly suggest) how a design would be adopted and used in everyday practice.

This essay is written to help counterbalance what we too often perceive as an unquestioning adoption of the doctrine of usability evaluation by user interface researchers and

Greenberg, S. and Buxton, B. (2007) Usability Evaluation Considered Harmful (Some of the Time). Report 2007-879-31, Department of Computer Science, University of Calgary, Calgary, AB, Canada. T2N 1N4. September.

practitioners. To set the scene, we first describe one of the key problems: how the push for usability evaluation in education, academia, and industry has led to the incorrect belief that designs – no matter what stage of development they are in – must undergo some type of usability evaluation if it is to be considered part of a successful user-centered process. Next, we illustrate how problems can arise by describing a variety of situations where evaluation is considered harmful. First, we argue that scientific evaluation methods do not necessarily imply good science. Second, we argue that premature evaluation of early designs can kill promising ideas or the pursuit of multiple competing ideas. Third, we argue that traditional user evaluation of designs that rely on cultural adoption would not provide meaningful information. All is not lost, for we then give general suggestions of what we can do about this. We close by pointing to others who have debated the merits of user evaluation within the CHI context.

THE HEAVY PUSH FOR USABILITY EVALUATION

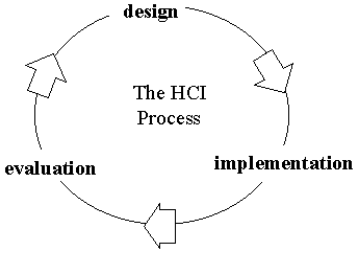
Usability evaluation is central to today's practice of HCI. In HCI education, it is a core component of what students are taught. In academia, evaluation of designs is considered the de facto standard for submitted papers to our top conferences. In industry, interface specialists regard evaluation as a major component of their work practice.

HCI Education

The ACM SIGCHI Curriculum for Human Computer Interaction formally defines HCI as

> "a discipline concerned with the design, *evaluation* and implementation of interactive computing systems for human use…" [15, emphasis added].

The curriculum stresses the teaching of evaluation methodologies as one of its major modules. This has certainly been taken up in practice, where the typical undergraduate HCI course stresses usability evaluation as a key course component in both lectures and student projects [7,12], e.g., by user testing, inspection, or controlled study. Following the ACM Curriculum, the canonical development process drummed into students' heads is the iterative process of design, implement, evaluate, redesign, re-implement, re-evaluate, and so on [7,12,15]. Because evaluation methodologies are easy to teach, learn, and examine (as compared to methods that teach people how to design), it has become perhaps the most concrete learning objective in a standard HCI course.



CHI Academic Output

Our key academic conferences such as ACM CHI, CSCW and even UIST strongly suggest that authors validate new designs of an interactive technology. For example, the ACM CHI 2008 *Guide to Successful Submissions* states:

> "does your contribution take the form of a design for a new interface, interaction technique or design tool? If so, you will probably want to demonstrate 'evaluation' validity, by subjecting your design to tests that demonstrate its effectiveness. [19]

The consequence is that the CHI academic culture generally accepts the doctrine that submitted papers on system design must include an evaluation – usually controlled experimentation or empirical usability testing – if it is to have a chance of success. Not only do authors believe this, but so do reviewers:

> "Reviewers often cite problems with validity, rather than with the contribution per se, as the reason to reject a paper" [19].

Our own combined five-decades of experiences intermittently serving as Program Committee member, Associate Chair, Program Chair or even Conference Chair of these and other HCI conferences confirm that this ethic – while sometimes challenged – is fundamental to how many papers are written and judged. Indeed, Barkhuus and Rode's analysis of ACM CHI papers published over the last 24 years found that the proportion of papers that include evaluation – particularly empirical evaluation – has increased substantially, to the point where almost all accepted papers have some evaluation component [1].

Industry

Over the last decade, industries are incorporating interface methodologies as part of their day-to-day development practice. This often includes the formation of an internal group of people dedicated to considering interface design as a first class citizen. These groups tend to specialize in usability evaluation. They may examine existing work practices, which in turn helps inform requirements gathering. They may evaluate different design approaches, which in turn leads to a judicious weighing of the pros and cons of each design. They may test interfaces under iterative development – paper prototypes, running prototypes, implemented sub-systems – where they would produce a prioritized list of usability problems that could be rectified in the next design iteration. This emphasis on evaluation is most obvious when interface groups are composed mostly of human factors professionals trained in rigorous evaluation methodologies.

Why this is a problem

In education, academia and industry, usability evaluation has become a critical and necessary component in the design process. Usability evaluation is core because it is truly beneficial in many situations. The problem is that academics and practitioners often blindly apply usability evaluation, which includes some situations where – as we will argue in the following sections – it gives meaningless or trivial results, and can misdirect or even kill future design directions.

EVALUATION AS WEAK SCIENCE

In this section, we emphasize concerns regarding how we do our evaluations. While we may use scientific methods to do our evaluation, this does not necessarily mean we are always doing good science or product development.

The Method Forms the Problem.

In the early days of CHI, a huge number of evaluation methods were developed for practitioners and academics to use. For example, John Gould's classic article *How to Design Usable Systems* is choc-full of pragmatic discount evaluation methodologies [11]. The mid-'80s to '90s also saw many good methods developed and formalized by CHI researchers: quantitative, qualitative, analytical, informal, contextual and so on (e.g., [20]). The general idea was to give practitioners a methodology toolbox, where they could choose a method that would help them best answer the problem they were investigating in a cost-effective manner. Yet Barkhuus and Rode note a disturbing trend in the recent ACM CHI publications [1]: evaluations are dominated by quantitative empirical evaluations (about 70%) followed by qualitative empirical evaluations (about 25%). As well, they report that papers about the evaluation methods themselves have almost disappeared. The implication is that ACM CHI now has a methodology bias, where certain kinds of methods are considered more 'correct' and thus acceptable than others.

The consequence is that people now likely generate 'problems' that are amenable to a chosen method, rather than the other way around. That is, they choose the method (perhaps the one they are more familiar with or that is perceived as 'favored' by review committees), and then find or fit a problem match it. Our own anecdotal experiences confirm this: a common statement by students is 'if we don't do a quantitative study, the chances of a paper getting in are small'. That is, they choose the method (e.g., controlled study) and then concoct a problem that fits that method. Alternately, they may emphasize aspects of an existing problem that lends itself to that method, where that aspect may not be the most important one that should be considered. Similarly, we noticed methodological biases in reviews, where papers using non-empirical methodologies are judged more stringently.

Existence Proofs.

Designs implemented in research laboratories are often conceptual ideas. They are usually intended to show an alternate way that something can be done. In these cases, the role of evaluation ideally shows that this alternate method is better – hopefully much better – than the existing 'control' method. Putting this into terms of hypothesis testing, the alternative (desired) hypothesis is (in very general terms): "When performing a series of tasks, the use of the new method leads to increased human performance when compared to the old method".

What most researchers then try to do – often without being aware of it – is to create a situation favorable to the new method. The implicit logic is that they should be able to demonstrate *at least one case* where the new method performs better than the old method; if they cannot, then this method is likely not worth pursuing. In other words, the evaluation is an existence proof.

This seems like science, for hypothesis formation and testing are at the core of the scientific method. Yet it is, at best, weak science. The scientific method advocates risky hypothesis testing: the more the test tries to refute the hypothesis, the more powerful it is. If the hypothesis holds in spite of attempts to refute it, there is more validity in its claims [27, see Chapter 9]. In contrast, the existence proof as used in HCI is confirmative hypothesis testing, for the evaluator is seeking confirmatory evidence. This safe test produces only weak validations of a method. Indeed, it would be surprising if the researcher could not come up with a single scenario where the new method would prove itself as being somehow better than an existing method.

The Lack of Replication.

Good science also demands replication, and the same should be true in CHI [13,27]. Replication serves several purposes. First, the HCI community should replicate evaluations to verify claimed results (in case of experimental flaws or fabrication). Second, the HCI community should replicate for more stringent and more risky hypothesis testing. While the original existence proof at least shows that an idea has some merit, follow-up tests are required to put bounds on it, i.e., to discover the limitations as well the strengths of the method [13,27].

The problem is that replications are not highly valued in CHI. They are difficult to publish (unless they are controversial), and are rarely considered a strong result. Again, dipping into experiences on program committees, the typical referee response is 'it has been done before; therefore there is little value added'. Industry is no better, as replication is likely considered an uneconomical use of resources.

What exasperates the "it has done before" problem is that this reasoning is applied even more heavily to breakthrough technologies. For many people, the newer the idea and the less familiar they are with it, the more likely they are to see other's explorations into its variations, details and nuances as the same thing. That is, the granularity of distinction for the unknown is incredibly coarse. For example, most reviewers are well versed in graphical user interfaces, and often find evaluations of the slight performance differences between (say) two types of radial menu as acceptable. However, reviewers considering an exploratory evaluation of (say) a new large interactive multi-touch surface, or of a tangible user interface almost inevitably produce the "it has been done before" review unless there is a grossly significant finding. Thus variation and replication in unknown areas must pass a higher bar if they are to be published.

All this leads to a dilemma in the HCI culture. We demand validation as a pre-requisite for publication or for continued product development, even though these first evaluations are typically confirmatory and thus weak. Yet we rebuff publication or pursuit of replications, even though they deliberately challenge and test prior claims and are thus stronger.

Objectivity *vs.* Subjectivity.

The attraction of quantitative empirical evaluations as our status quo (the 70% of our CHI papers as reported in [1]) is that it lets us escape the apparently subjective: instead of expressing opinions, we have methods that give us something that appears to be scientific and factual. Even our qualitative methods (the other 30%) are based on the factual: they produce descriptions and observations that bind and direct the observer's interpretations. The challenge, however, is the converse. Our factual methods do not respect the subjective: they do not provide room for the experience of the advocate, much less their arguments or reflections or intuitions about a design.

The argument of objectivity over subjectivity has already been considered in other design disciplines, with perhaps the best discussion found in Snodgrass and Coyne's discussion of design evaluation in architecture by the experienced designer-as-assessor [23]:

> [Design evaluation] is not haphazard because the assessor has acquired a tacit understanding of design value and how it is assessed, a complex set of tacit norms, processes, criteria and procedural rules, forming part of a practical know-how. From the time of their first 'crit', design students are absorbing design values and learning how the assessment process works; by the time they graduate, this learning has become tacit understanding, something that every practitioner implicitly understands more or less well. An absence of defined criteria and procedural rules does not, therefore, give free rein to merely individual responses, since these have already been structured within the framework of what is taken as significant and valid by the design community. An absence of objectivity does not result in uncontrolled license, since the assessor is conforming to unspoken rules that, more or less unconsciously, constrain interpretation and evaluation. If not so constrained, the assessor would not be a member of the hermeneutical community, and would therefore have no authority to act as an assessor. (p.123)

One way to recast this is to propose that the subjective arguments, opinions and reflections of those deemed expert by the community should be considered just as legitimate as results derived from our more objective methods. Using a different calculus does not mean that one cannot obtain equally valid but different results. Our concern is that the narrowing of the calculus to essentially one methodological approach is negatively narrowing our view and our perspective, and therefore our potential contribution to CHI.

A final thought before moving on. We have one methodology, art and design schools have another. Is anyone surprised that art and design are remarkable for their creativity and innovation? While we remark on our rigor, we also bemoan the lack of design and innovation. Could there be a correlation between methodology and results?

EVALUATION AS DAMAGING TO DESIGN

In this section, we emphasize several common situations where evaluation, if done prematurely, not only adds little value, but can actually kill what could have been promising design idea. The basic question is whether we should even be evaluating some of our designs.

Sketches vs. Prototypes

Early designs are best considered as sketches. They illustrate the essence of an idea, but have many rough and/or undeveloped aspects to it. When an early design is displayed as a sketch, the team recognizes it as something to be worked on and developed further. Yet early designs can also be implemented and maintain their sketch-like properties, as yet another way to explore the ideas behind highly interactive systems. When systems are created as interactive sketches, they serve as a vehicle that helps a designer make vague ideas concrete, reflect on possible problems and uses, discover alternate new ideas and refine current ones [3,26,28,29].

The problem is that these working interactive sketches – especially when their representation conveys a degree of refinement beyond their intended purpose – are often mistaken for prototypes. Indeed, the HCI literature rarely talks about working systems as a sketch, and instead elevates them to low / medium / high fidelity prototyping status [3]. Yet the reality is that these are all inappropriate terms. There is no high, medium or low fidelity. There is only fidelity that is appropriate and inappropriate to the purpose of the model or implementation. At the extremes, prototypes are very different in purpose from a sketch. Buxton [3] characterizes the differences as defining the extremes along several continua, as illustrated in Figure 1.

By definition, a sketch – even if implemented as an interactive system – is a roughed out design. It will have

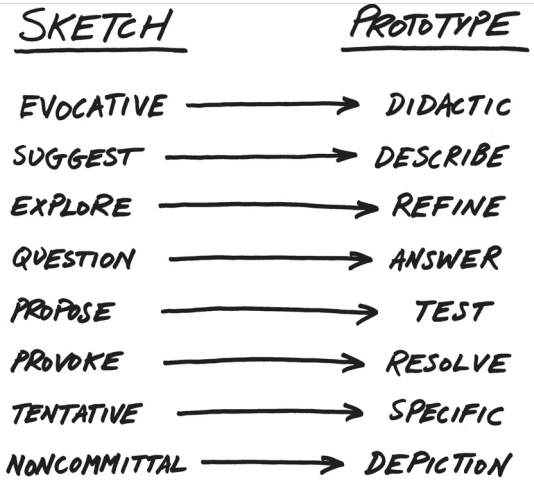


Figure 1. The sketch to prototype continuum, from [3] with permission.

many holes, deficiencies, and undeveloped attributes. In contrast, a prototype aids idea evaluation, either by validating it with clients as they try it out, or through usability testing. Consequently, premature testing of the sketch as prototype could, unsurprisingly, find significant problems that could kill the design outright, especially if a novel design is compared to one that is more conservative.

Getting the Right Design *vs.* Getting the Design Right

On the flip side, evaluation of sketches may also encourage developers to solve any of the problems seen by iterative refinement, as this is what the 'design, implement, evaluate' life cycle advocates. This leads to local hill climbing, where much effort is expended in 'Getting the Design Right' (Figure 2a). Unfortunately, evaluation of early sketches is often at the expense of considering and / or developing other ideas.

A sketch typically illustrates only one of many possible designs and variations under consideration. Early design demands many idea sketches, reflecting on this multitude of competing ideas, and choosing the one(s) that appear the most promising (Figure 2b). The promising idea is then further varied and developed until it can serve as a testable prototype. That is, sketching is about 'Getting the Right Design' [29,3]. Only afterwards does one work on 'Getting the Design Right' of a particular idea through iterative testing and development. Thus sketching is akin to a heuristic that helps one move closer to the global maxima by circumventing the local hill climbing problem.

The particular methodologies favored within CHI confound this problem. Most, like think-aloud observations, task centered walkthroughs, and heuristic evaluation, tend to focus on the negative: Where are the problems? What are the bugs? However, they do not inform us about the benefits. Yet ultimately, the underpinning of a meaningful evaluation is a cost/benefit analysis. The problem with our methodologies is that cost (problems) is easier to measure than benefit, and this focus on problems risks biasing decisions far too early in the process. Some of our early buggy designs may actually be the one that has the most potential for benefit in the long run, but we have no way of knowing this. The net result is that we eliminate ideas too early, we consider far too few ideas at all, we converge on that which we can measure, which is almost always that which we are already familiar with, and innovation degrades into a refinement of the known rather than the initiation along new trajectories.

More generally, the ACM curriculum [15] suggests that design is an equal partner with evaluation. Yet the standard iterative cycle promoted within the CHI community is counter to traditional design practice. At issue is how the design/implement/test loop encourages the sequential evolution/refinement of ideas, rather than the multiple parallel solutions that characterize most traditional design disciplines [3]. We include ourselves in this criticism, as we have authored one of the first papers advocating an iterative approach to design [31], and have advocated it in education [31].

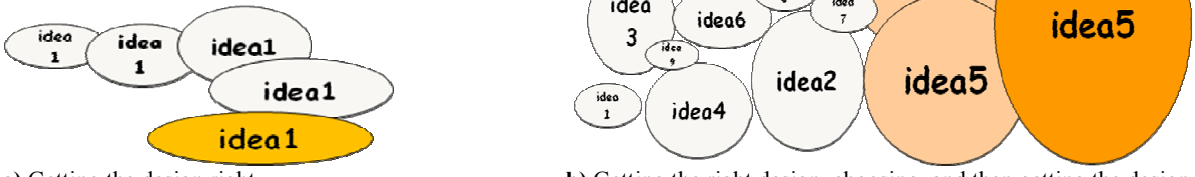
EVALUATION IGNORES CULTURAL ADOPTION & USE

Except for ethnographic and field studies (more commonly found in CSCW *vs.* CHI), evaluation methods tend to consider products outside of their cultural context. Yet the reality is that cultural uptake of a product is hugely significant in terms of how it is actually used, and how that influences product evolution over time.

Usable or Useful?

Thomas Landauer eloquently argues in his book *The Trouble with Computers* that most computer systems are problematic because not only are they are too hard to use, but they do too little that is useful [17]. Most evaluation methodologies target the 'too hard to use' side of things: system designs are proven effective ('easy to use') when activities can be completed with minimal disruptive errors, efficient when tasks can be completed in reasonable time and effort, and satisfying when people are reasonably happy with the process and outcome [14,17]. The problem is that these measures do not indicate the Landauer's second concern: design usefulness.

This distinction between usability and usefulness is not subtle. Indeed, the technological landscape is littered with unsold products that are highly usable, but totally useless. Conversely – and often to the chagrin of usability professionals – many product innovations succeed because they are very useful or if they become fashionable: they sell



a) Getting the design right

b) Getting the right design, choosing, and then getting the design right

Figure 2. Sketching first, iterative design and evaluation later

even though they have quite serious usability problems. In practice, good usability in successful products often happens after – not before – usefulness. A novel and useful product (even though it may be hard to use) is taken up by people, and then competition over time forces that product (or a competitor's product) to evolve into something that is more usable. The world wide web is proof of this: many early web systems were abysmal but still highly used (e.g., airline reservations systems): usability came somewhat late in the game. The violin is an even more obvious example.

However, usefulness is a very difficult thing to evaluate, especially given our standard methodologies in CHI. In most cases, it is often evaluated indirectly by determining (perhaps through a requirements analysis) what tasks are important to people, and using those tasks to seed usability studies. If people are satisfied with how they do these tasks, then presumably the system will be both usable and useful. Yet determining usefulness of new designs is hard. In the Innovator's Dilemma, Christensen [5] argues that customers (and by extension designers who listen to them) often do not understand how new innovative technologies – especially those that seem to under-perform compared to existing counterparts – can prove useful to them. It is left to 'upstart' companies to develop these technologies: they find usage niches where they prove highly useful, and redesign them until they later (sometimes much later) become highly useful – and usable – in the broader context.

Yet usability evaluation is predisposed to the world changing by gradual evolution; iterative refinement will produce more usable systems, but not radically new ones. It is ill disposed towards discontinuities as suggested by the Innovator's Dilemma, where sudden uptakes of useful or fashionable technologies by the community occur. If all we do is usability evaluation (which in turn favors iterative refinement), we can have a modest impact by making existing things better. What we cannot do is have major impact without creating new innovations .

Initial Idea Usability vs. Normative Cultural Adoption.

Usability testing – whether via usability studies, inspection, or controlled experimentation – has many known benefits as practiced today. In almost all cases, they identify usability bugs that hinder or stop a person from continuing their task, or they produce performance measures that suggest how long it takes a person to perform a task, or they compare how one person uses different systems / methods. These tests are appropriate for settings with well-known tasks and outcomes that determine how useful a system will be. Unfortunately, they completely fail to consider how a system – particularly if it is novel – will be adopted by a culture over time. Let us consider a few important examples to place this into context.

Marconi's wireless radio. In 1901, Guglielmo Marconi conducted the first trans-Atlantic test of wireless radio, where he transmitted in Morse code the three clicks making up the letter 's' between the United Kingdom and Canada[1]. If considered as a usability test, it is less than impressive. First, the equipment setup was onerous: he even had to use balloons to lift the antenna as high as possible. Second, the sound quality was barely audible. He wrote

> "I heard, faintly but distinctly, pip-pip-pip. I handed the phone to Kemp: "Can you hear anything?" I asked. 'Yes,' he said. 'The letter S.' He could hear it."

Yet, his claim was controversial, with many scientists believing that the clicks were produced by random atmospheric noise mistaken for a signal[2].

Whether or not this 'usability test' demonstrated the feasibility of wireless transmission, it is as interesting to consider how Marconi's vision of how it would be used changed dramatically over time. Marconi is purported to have envisioned radio as a means for maritime communication between ships and shore; indeed, this was one of its first uses. He did not foresee what we take as commonplace: broadcast radio. If Marconi tried to publish at CHI or pass this through a usability team at a company, his system's un-usability and (perhaps) his limited vision would likely have led to immediate rejection.

While one could argue back that wireless radio is more of an infrastructure capability than an end user system (and thus outside the scope of usability tests), consider the automobile as an innovation. Cars were designed with end users in mind. Yet early ones were expensive, noisy, unreliable. They demanded considerable expertise to maintain and drive them. They were also impractical, as there was little in the way of infrastructure to permit regular travel. Could the early car have passed the CHI criteria?

Bush's Memex. Let us now consider how several great innovations in Computer Science would have fared. In 1945, Vannevar Bush introduced the idea of cross-linked information in his seminal article *As We May Think* [2], which in turn inspired the fields of Hypertext and, of course, the World Wide Web. Bush described a system called 'Memex' based on linked microfilm records. Yet he never built a system, let alone evaluated it. We can imagine how an article of similar tone would fair if submitted to a CHI review process:

> "Bush proposes an interesting idea. Yet the suggested design is unimplemented and untested. Microfilm is likely too unwieldy to be practical. The work is premature. We recommend that he resubmit after he builds and evaluates his design."

Of course, Bush's vision wasn't even correct. His vision was constrained to knowledge workers; he certainly never anticipated the use of linked records for what is now the mainstay of the web: social networking, e-commerce, pornography and gambling. Thus even if he had conducted a usability test, they would have been based on tasks not

[1] http:/www.pbs.org/wgbh/aso/databank/entries/dt01ma.html,

[2] http://en.wikipedia.org/wiki/Guglielmo_Marconi

considered central to today's culture. Still, there is no question that this was a valuable idea with profound influence on how people considered and eventually developed a new technology.

Sutherland's Sketchpad. In 1963, Ivan Sutherland produced Sketchpad, perhaps the most influential system in computer graphics and CAD [25]. Sketchpad was an impressive object-oriented graphics editor, where operators manipulated a plethora of physical controls (buttons, switches, knobs) in tandem with a light pen to create a drawing. Again, no evaluation was done. Even if it were, it would probably have fared poorly due to the complexity of the controls and the poor quality display, something that disappeared only when technology caught up. Again, let us imagine a CHI review.

> "This untested system, while promising, is likely far too complex for the simple designs produced from it. Due to the number of controls and the difficulty of using the light pen, the level of operator training would be very costly, especially given the simplicity of what one could do with it. At the very least, we need a baseline measure of Sketchpad's efficacy. We recommend that the author test Sketchpad use against a trained draftsman working with paper and pencil while trying to produce a variety of drawings".

Engelbart's NLS. In 1968, Douglas Engelbart gave what is arguably the most important system demonstration ever held in Computer Science [9]. He and his team showed off the capabilities of his NLS system. His vision as realized by NLS had a profound influence on graphical interfaces, hypertext, and computer supported cooperative work. Yet Engelbart's vision was about enhancing human intellect rather than ease of use. He believed that highly trained white collar people would use systems such as his to push the envelope of what is possible. Again, we could imagine a CHI review.

> "While Engelbart impressively demonstrates a working system, it has only been used by himself and his team. Thus, it cannot be considered an objective validation of its capabilities. As well, his system demands such a high degree of training that it would likely fail any laboratory test of casual users."

Engelbart also failed to envision or predict the cultural adoption of his technologies by everyday folks for mundane purposes: he was too narrowly focused on productive office workers. Similar to Memex, even if Engelbart had done usability tests, they would have been based on a user audience and set of tasks that do not encompass today's culture.

There are several points to make about the above examples.

1. The systems – whether ideas or working ones – were instances of a vision of what computer interfaces could be like. It is the vision as well as the technology that was critical.
2. The visions foresaw the creation of a new culture of use, where people would fundamentally change what they would be able to do.
3. None of the systems were immediately realized as products. Indeed, there was a significant lapse of time before the ideas within them were taken up by others in new systems.
4. The way the ideas were taken up both in systems and by culture evolved considerably from the original vision of use: even our best visionaries had problems predicting how cultures would adopt technologies to their personal needs. Yet the vision was critical for stimulating work in the area.
5. Even if usability tests had occurred, they likely would have been meaningless. They would have been based around user groups and task sets that would have little actual correspondence to how the product would evolve in terms of its audience and actual uses.

The key point is that even great ideas are rarely 'usable' or even testable when they are first formed. As with the above examples, the technology may not be ready or may be overly costly. The actual idea itself likely has significant rough edges or limited capabilities that would undermine its immediate usability and usefulness. The culture of users may not be ready for it, and indeed may not be ready for many years. The expected uses may not fit with the actual uses that would evolve.

Today's Compelling Ideas.

To put this into today's perspective, we are now seeing many compelling ideas that suffer from the similar limitations of the historic ones mentioned above. For example, consider the challenges of creating Ubiquitous Computing technologies (Ubicomp) for the home. Often, the technology required is too expensive (e.g., powerful tablet computers), the infrastructure is not in place (e.g., configuring hardware and wireless networks), the necessary information utilities are unavailable, or the system administration is too hard [8]. Even more important, the culture is not yet in place to exploit such technology: it is not a case of one person using Ubicomp, but of a critical mass of home inhabitants, relations, and friends adopting and using that technology. Only then does it become valuable. Again, cultural and technical readiness is needed before a system can be deemed 'successful'. How does one evaluate that except after the fact? Do toy deployments and evaluations really test much of interest?[3]

As a counterpoint, consider the many highly successful social systems now available on the web: Youtube, Facebook, Myspace, Instant Messaging, SMS, and so on. From an interface and functionality perspective, most of these systems can be considered quite primitive. Indeed, we can even criticize them for their poor usability. For example, *Youtube* does not allow one to save a viewed video, so that one must be online to view a previously seen one; one must buffer the entire video even though only a fragment at the end is wanted, one must wait. Yet usability

[3] They do, but as a way to inform design critique and reflection rather than testing. This interpretative use of testing is not normally considered part of our traditional process.

issues such as these are minor in terms of the way the culture has found a technology useful, and how the culture adopted and evolved its use of such systems.

Our argument is that using standard evaluation methods to validate such systems outside its culture of use is almost pointless (excepting for identifying slight usability problems). This leads to a dilemma: how can we create what could become culturally significant systems if we demand that the system be validated before a culture is formed around it? Indeed, this dilemma leads to a major frustration within CHI. We predominantly produce technology that is somewhat better than its antecedents, but these technologies rarely make it into the commercial world (although they may influence it somewhat). We also see technologies that are highly successful but not developed by the CHI community, so we are left to evaluate its usability only after the fact. There is something wrong with this picture.

WHAT TO DO

We have argued that evaluation – while undeniably useful in many situations – can be harmful if applied inappropriately. There are several initiatives that we as a community can do to remedy this situation.

First, we need to recognize that user centered design is not equal to usability evaluation. There are many aspects of user-centered design: understanding requirements, considering cultural aspects, developing and showing clients design alternatives, and so on. Usability evaluation is just one component in a user-centered design process, and as such should be used only when appropriate.

Second, we need to judge whether evaluation at a particular point in our design cycle would produce anything meaningful. This means we need to continually reflect on our process, and consider the pros and cons. If the answer is 'no', then we should not do an evaluation. We outlined above several situations where this is likely: in very early design stages, in cases where usefulness overshadows usability, in instances where cultural uptake dominates how a system will actually be used.

Third, as a community we need to stop this blanket insistence on interface evaluation. This is not to say that we should accept hand-waving and slick demos as an alternative. As both an academic and practitioner community, we need to recognize that there are many appropriate ways to validate one's work, i.e., not just by a formal evaluation, but by design rationale, by visions of what could be, by expected scenarios of use, by reflections, by case studies, by participatory critique, and so on. At a minimum, authors should articulate why things were done, what else was considered, what they learned, how it fits in the broader context of both prior art and situated context, what next, and why it might be worth noting. These are all criteria that would be expected in any respected design school or firm. There is a rigour. There is a discipline. It is just not the same rigour and discipline that we currently encourage, teach, practice or accept. Academic paper submissions or product descriptions should be judged by the type of system or situation that is being described and whether the method the inventors used to argue its merits and weaknesses are reasonable.

Fourth, when user evaluations are appropriate for validating our designs, we should recognize that the formulatic way we do our evaluations (or judge them as publishable) often results in weak science. We need to change our methods to produce strong science. For really novel systems, existence proofs are likely appropriate. For mainstream systems or slight variations of established ideas, we should likely favor risky hypothesis testing. We certainly should be doing more to help others replicate our results (e.g., by publishing data and/or making software available), and we should be more open-minded about accepting and encouraging replications in our literature.

Fifth, we should look to other disciplines to consider how they judge design worthiness. For example, the practice of design as taught in disciplines such as architecture and industrial design employs the notion of a design studio: a place where people develop ideas into artifacts, and where surrounding people are expected to engage in discussion about these artifacts as they are being formed. These fields recognize that early designs are just 'sketches' that illustrate an idea in flux. Sketches are meant to change over time, and active discussion can influence how they change. Early evaluation is usually through the Design Critique (or 'Crit'). The designer presents the artifact to the group (typically a mix of senior and junior people), and explains why the design has unfolded the way it has. Members of the group respond: by articulating what they like and dislike about the idea, by challenging the designer's assumptions through a series of probes and questions, and by offering concrete suggestions of ways to improve the design. This is a reflective and highly interactive process: constructive criticisms and probing demands that designer and criticizers alike develop and share a deep understanding of the design idea and how it interacts within its context of use.[4] Similarly, we need to understand methods that evaluate cultural aspects of designs. We are seeing some of this at ACM CSCW, where ethnographic approaches are now considered the norm. Other fields, such as Communication and Culture, have their own methods that may be appropriated for our use. While the ACM CHI Conference has recently encompassed design, it is typically as a separate track outside the traditional mainstream.

Sixth, if we are going to incorporate the "crit" into our practice, then we need to make sure that it is informed, respectful, open and constructive. In terms of being informed, among other things, this implies a far stronger literacy in our history than is currently the case. Innovation

[4] http://www.scottberkun.com/essays/essay23.htm

and science is not just invention. It involves scholarship and theory, as well as creativity – something that is too often missed on both sides of the design *vs* science divide. And, as for the social process of the crit, this is like any other skill – it requires constant practice to nurture to the point of maturity. That is why it becomes part of the routine from one's first day in art and design school. None of this happens by magic. What we are arguing for is a change in culture and in how we train our professionals [31], not a mere technique.

RELATED WORK

We are not the first to raise cautions about the doctrine of evaluations in CHI.

Henry Lieberman seeded the debate in his *Tyranny of Evaluation*, where he damns usability evaluation and the insistence that CHI places on it [18]. His position was challenged by Shumin Zhai, who essentially says that in spite of the concerns, our evaluation methods are better than doing nothing at all [30]. Cockton continued the debate in 2007 [6], where he argues that the problem is not whether one should do evaluations, but that there is a lack of methods that are useful to various design stages, or to various practitioners (e.g., inventor *vs*. artist *vs*. designer *vs*. optimizer). Dan Olsen is moderating a panel on evaluating interface systems research at UIST 2007. He raises concerns about how our expectations on measures of usability when evaluating interactive systems can create a usability trap, and offers other criterion for helping us judge systems [21].

Others raise concerns about the methods we use. Many standard textbooks offer the standard caveats to empirical testing, e.g., internal vs. external validity, statistical vs. practical significance, generalization, and so on [7]. Narrowing to the CHI arena, Stanley Dicks argues on the uses and misuses of usability testing in [24], while Barkhuus and Rode analyze the preponderance of evaluation testing in CHI and raise concerns about how evaluations are now typically done [1]. Kaye and Sengers look at the evolution of evaluation in CHI: they stress the 'Damaged Merchandise' controversy that had practitioners from different fields challenging the usefulness of methods, particularly between advocates of discount methods *vs.* formal quantitative methods [16]. Pinelle and Gutwin analyzed evaluation in CSCW from 1990-1998 and found that almost 1/3 of the systems were not formally evaluated, but perhaps more importantly that only about ¼ included evaluation in a real-world setting [22]. Given that CSCW systems are often cultural, this raises serious questions about the evaluations that ignore real world context.

Of course, there are many people who consider the contributions of other non-evaluation methods to design. For example, Tohidi et. al. consider sketches as an effective way of getting reflective user feedback [28], while Buxton more generally considers the role of sketching in the design process [3]. On the cultural side, Gaver et. al. are developing methods that probe cultural reactions and technology uptake by niche cultures [10].

Finally, several splinter groups within the CHI umbrella arose in part as a reaction to evaluation expectations. ACM UIST emphasizes novel systems, interaction techniques, and algorithms – while evaluation is desired, it is not required if the design is inspiring and well argued (although they too are debating about how they are falling in the evaluation trap [21]). In its early years, ACM CSCW incorporated and nurtured ethnography and qualitative methods as part of its methodology corpus. ACM DIS favors works that emphasizes design and the design process over evaluation.

CONCLUSION.

As mentioned in the beginning of this paper, we do not want our message mis-interpreted. We do not want to hear others say that 'the authors argued that we no longer have to do usability evaluation'. We are advocates of usability evaluation, and have used it consistently as part of our own work. What we argue is that we should apply usability evaluation consciously. We should decide if it is appropriate to the situation we are examining and, if so, we should chose a method that truly informs us about that situation. We should be open to other non-empirical methods - design critiques, design alternatives, case studies, cultural probes, reflection, design rationale - as being perhaps more appropriate for some of our situations.

It would be just as inappropriate to drop usability testing altogether in favor of the approaches that we are advocating. Zhai argues that usability evaluation is the best game in town [30], and we qualify by saying that this is true in most, but not all cases. For some, other methods (some which are listed above) are more appropriate. However, in all cases a combination of methods – from empirical to non-empirical to reflective – will likely help triangulate and enrich the discussion of a system's validity. It is just a matter of balance, but then, that is the true essence of evaluation anyhow!

While some of our concerns may appear novel to young CHI practitioners, those who have been around will have heard them before and will likely have their own opinion. Regardless of who you are, consider how you can help enrich CHI. Join the debate. Change your development practices as a researcher and practitioner. Reconsider how you judge papers and systems. Teach our new professionals that HCI ≠ Usability Evaluation; it is far more than that.

ACKNOWLEDGMENTS

This essay assimilates the many opinions, discussions, and arguments we have had or overheard between CHI colleagues over many, many years (mostly in bars, restaurants and hallway conversations). We thank them collectively.

REFERENCES

1. Barkhuus, L., Rode, J. From Mice to Men – 24 Years of Evaluation in CHI. *ACM CHI'07 – Alt.CHI.* http://www.viktoria.se/altchi/ (2007).
2. Bush, V. As We May Think. *Atlantic Monthly*, July. (1945)
3. Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design.* Morgan Kaufmann, (2007).
4. Buxton, W. & Sniderman, R. Iteration in the Design of the Human-Computer Interface. *Proc 13th Ann. Meeting, Human Factors Assoc. of Canada,* (1980), 72-81.
5. Christensen, C. *The Innovator's Dilemma.* Harper Business Essentials. Collins. (2003).
6. Cockton Make Evaluation Poverty History. A*CM CHI'07 – Alt.CHI.* http://www.viktoria.se/altchi/. (2007)
7. Dix, A., Finlay, J., Abowd, G. and Beale, R. *Human Computer Interaction, 2nd Edition*, Prentice Hall, (1993)
8. Edwards, W. and Grinter, R. At Home with Ubiquitous Computing: Seven Challenges. *Proc Conf Ubiquitous Computing.* Lecture Notes In Computer Science, vol. 2201. Springer-Verlag, London, (2001), 256-272.
9. Engelbart, D.C. and English, W.K. A Research Center for Augmenting Human Intellect. *AFIPS Fall Joint Computer Conference,* Vol. 33, (1968), 395-410.
10. Gaver, B., Dunne, T., and Pacenti, E. 1999. Design: Cultural probes. *ACM Interactions* 6:1, (1999), 21-29.
11. Gould, J.D. How to design usable systems. in R. Baecker, J. Grudin, W. Buxton and S. Greenberg (eds) *Readings in Human Computer Interaction: Towards the Year 2000*, Morgan-Kaufmann, (1996), 93-121.
12. Greenberg, S.. Teaching Human Computer Interaction to Programmers. 3(4), *ACM Interactions*, (1996), 62-76.
13. Greenberg, S. and Thimbleby, H. *The weak science of human-computer interaction*. *Proc CHI '92 Research Symposium on HCI* (1992)
14. Gutwin, C. and Greenberg, S. The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *Proc 9th IEEE Int'l Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'00).* (2000).
15. Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong and Verplank (1996) *ACM SIGCHI Curricula for Human-Computer Interaction.* Last updated 2004-06-03. http://sigchi.org/cdg/index.html
16. Kaye, J. and Sengers, P. The Evolution of Evaluation. *ACM CHI'07 – Alt.CHI.* http://www.viktoria.se/altchi/
17. Landauer, T. *The Trouble with Computers: Usefulness, Usability, and Productivity.* Cambridge, MA: MIT Press. (1995)
18. Lieberman, H. The Tyranny of Evaluation. web.media.mit.edu/~lieber/Misc/TyrannyEvaluation.html, *ACM CHI Fringe,* (2003).
19. Newman, W. *CHI Guide to a Successful Archive Submission. http:*//www.chi2008.org/archiveGuide.html, (2008).
20. Nielsen, J. *Usability Engineering.* Morgan Kaufmann. (1993).
21. Olsen Jr., D. (2007, in press). Evaluating User Interface Systems Research. *Proc ACM UIST'07*. ACM Press.
22. Pinelle, D. and Gutwin, C. A Review of Groupware Evaluations. *Proc 9th IEEE Int'l Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'00).* (2000). 86-91.
23. Snodgrass, A. & Coyne, R. *Interpretation in architecture: Design as a way of thinking.* London: Routledge. (2006).
24. Stanley Dicks, R. Mis-Usability: On the Uses and Misuses of Usability Testing. *Proc ACM SIGDOC*, (2002)
25. Sutherland, I. *Sketchpad: A man-machine graphical communication system.* PhD Thesis, MIT, (1963).
26. Suwa and Tverskey. External representations contribute to the dynamic construction of ideas. In M. Hegarty, B. Meyer, and N. H. Narayanan (Eds.), *Diagrams* NY: Springer-Verlag, (2002), 341-343.
27. Thimbleby, H. *User Interface Design.* ACM Press Frontier Series, Addison-Wesley, (1990).
28. Tohidi, M., Buxton, W., Baecker, R., and Sellen, A. User Sketches: A Quick, Inexpensive, and Effective way to Elicit More Reflective User Feedback. *Proc. NordiCHI* (2006), 105-114.
29. Tohidi, M., Buxton, W., Baecker, R. and Sellen, A. Getting the Right Design and the Design Right: Testing Many is Better than One. *Proc ACM CHI,* (2006), 1243-1252.
30. Zhai, S. Evaluation is the worst form of HCI research except all those other forms that have been tried, essay published at *CHI Place*, (2003). http://www.almaden.ibm.com/u/zhai/publications.html
31. Anon. Placeholder for references kept anonymous for the blind review process.