

Department of Computer Science

Discount Usability Testing

Abstract

Usability testing need not be costly and time-consuming to be effective. Discount usability testing provides a set of cheap, fast and easy to apply techniques including think aloud, card sorting, walkthroughs, scenarios and heuristic evaluation. These methods can reveal major and minor usability issues in a system at a low cost. Although it is an advantage, evaluators need not be usability experts. These methods can be utilized early in the design cycle or during the implementation phase. In an Agile setting, discount usability testing works effectively because of the iterative nature of this model. Discount methods tend to find superficial problems; thus they usually are not suitable for in-depth usability studies. Discount usability testing is not a replacement of traditional usability testing, but can be very advantageous compared to doing nothing.

By: Yaser Ghanam

Revised by: Frank Maurer

Table of Contents

Table of Contents	2
Table of Figures	2
Introduction	3
Discount Usability Testing Methodologies	4
1. Think Aloud	4
2. Card Sorting	4
3. Walkthroughs	5
4. Scenarios	5
5. Heuristic Evaluation	6
Discount Usability in Agile Methods	8
Possible Approach	8
Methods	9
Advantages 1	1
Disadvantages 1	1
Conclusion 1	2
References 1	3

Table of Figures

Figure 1 - Task Centered Walkthrough Procedure according to [14]	. 5
Figure 2 – Scenario, obtained from [7]	. 6
Figure 3 - Usability Problems Found vs. Number of Evaluators, copied from [18]	. 7
Figure 4 – Agile development process with usability testing, obtained from [21]	. 9

Introduction

Usability engineering is the process of evaluating how usable a product is in order to improve the usability of the product and enhance the process by which products will be designed and developed in the future [1]. While some software products provide a novel and useful set of functionalities, some other products compete in the market by enhancing the usability of existing functionalities. Investing in improving the usability of an existing, or functionally ready-to-release, product can be justified by the fact that poor usability can easily coerce users to give up on a product and look for alternative solutions [2].

In the late 1990s, IBM introduced the concept of real objects. They tried to develop a set of applications in which the interface was supposed to look real and interact with the user just like what would happen in real life. The most famous application was the IBM RealPhone - an application that allows users to make phone calls from their computers. At the first glance, this seemed to be a ground breaking idea, especially with IBM promoting the product with slogans like "Welcome to the future; one without distracting windows and menu bars.", "If you can use a telephone, you can use this software", "Novice users can use it immediately" ... etc [3]. With all these big promises being publicized, it was very embarrassing for IBM to find out very shortly after releasing the product that their RealPhone ranked first in the "Interface Hall of Shame" [4]. It was found to be "violating nearly every aspect of proper interface design." It is not questionable that the objective of this project was not to provide a novel feature but to improve an existing one. However, it was a big failure because, as found later, there were absolutely no usability tests before the product was publicized.

While no one explicitly denied the benefits of conducting usability tests prior to releasing products, many did not adopt it due to the commonly perceived fact that it was expensive and time-consuming. In attempt to correct this perception, Nielsen and other usability practitioners coined the term "Discount Usability" in the early 1990s [5]. By introducing low-cost and easily accessible usability testing methodologies that value observation and interpretation over complex statistics, and value flexibility of procedure, space and time over expensive test labs and sophisticated experimentation, a new perception of usability engineering has emerged. According to Curtis as quoted in [6], usability engineering is no longer a practice of only multimillion dollar companies with very well-equipped test labs nor is it a huge burden on the project life cycle. But rather, it can be a cheap, fast and highly beneficial step in the development process for projects of any scale. Nielsen asserts that usability does not have to slow down your project or be complex or expensive to be effective [6].

This paper presents an overview of some discount usability techniques and discusses a framework to utilize these techniques in an agile environment. The paper also discusses the advantages and shortcomings of discount usability engineering suggesting contexts in which they can be beneficial and contexts where they better be avoided.

Discount Usability Testing Methodologies

Many lightweight, easy to learn and fast to conduct usability testing techniques have been proposed by usability experts. In order to cheaply find usability problems in a system, Nielsen and others proposed and applied techniques like card sorting, walkthroughs, scenarios, think aloud and heuristic evaluation [6]. This section briefly discusses each of these methods.

1. Think Aloud

Think aloud testing is the most popular technique for usability evaluation [8] developed at IBM by Clayton Lewis [9]. As the name suggests, this method relies on the user continuously speaking out his mind, aka thinking aloud, while performing a typical scenario in the system [10]. Users are asked to verbalize whatever they are planning to do, how they are trying to approach it, how they actually are doing it, and how they are feeling about it [11]. This allows usability evaluators to track the way the user constructs and arranges his mental models about the system in usage. The user may then be debriefed to obtain a more in-depth description of the cognitive activities associated with the tasks the user performed. Usability experts compare the user-generated mental models to those actually intended in the design of the system to check if there are any discrepancies.

A think aloud testing session is typically preceded by preparing scenarios that typical users of the system are likely to perform. Then, a group of users are asked to perform these scenarios and verbalize whatever they are trying to do and explain why they are going about it the way they are. This session is often audio or video taped. The recordings are to be reviewed by usability experts and probably the system developers in order to identify the usability issues.

To encourage participants to speak out their minds, a technique called co-discovery may be used where participants are asked to work the problem together, preferably in pairs. This way, they are communicating for their own benefit as opposed to a single participant who is asked to think aloud for the benefit of the test facilitator and thus they tend to verbalize their thoughts more [12].

2. Card Sorting

Another technique to uncover the user's mental model of an information space is card sorting. According to [6], card sorting is especially beneficial for information distribution or feature grouping purposes. By knowing how the user mentally groups commands, icons, and pieces of information, developers will be able to provide a more usable interface that meets the users' expectations.

If the developers want to group commands in menus or panels, and they are interested in finding out a convenient grouping scheme, the first step will be to have every command written on a separate index card with an optional one line description. The cards are then scattered on a table. Potential users of the system are asked to individually attempt to group the cards into piles according to similarity. Although the number of items in each pile should not be specified by the testers, they may want to encourage the users to produce a reasonably sized pile that is not too small or too large. Having all the commands grouped into different piles, the users are asked to group the piles into larger groups and give each of these large groups a unique name. Typically, the user needs about 30 minutes to complete the whole process of card sorting.

3. Walkthroughs

This is a fairly new usability technique that was developed in the early 1990s [13]. Although cognitive walkthroughs, persona walkthroughs and task-centered walkthroughs amongst some other types can all be used for usability testing, task-centered walkthroughs proved to be a highly practical discount usability method [14]. According to [15], the first thing to do is define the targeted audience and develop concrete descriptions of real-life tasks this audience is likely to perform. These descriptions help the designers articulate the set of features the system should provide to its potential users. Taking into consideration the background of the previously defined audience, and the amount of knowledge required to perform the fore described tasks, the first prototype of the interface can be designed. During and after the design phase, the prototype should be repeatedly evaluated on how well it supports the expected stories of the potential users. This evaluation can be done according to the procedure in Figure 1**Error! Reference source not found.** originally developed by Lewis and Rheiman [14].

```
SELECT one of the defined tasks and a potential user
FOR EACH user's action WHILE performing the task:
    IF the user:
    Cannot anticipate what is to be done next
    OR
    Does not seem to have sufficient knowledge to do it
    THEN
        There is a problem in the interface at this point
        Note the problem and any comments or solutions
        Tag the problem as repaired
```

Figure 1 - Task Centered Walkthrough Procedure according to [14]

4. Scenarios

High fidelity prototypes can be classified as vertical prototypes and horizontal prototypes. On the one hand, horizontal prototypes represent a user interface layer with a shallow implementation to give an overall idea about the system as a whole. On the other hand, vertical prototypes represent a reduced subset of fully implemented features that may not be completely integrated with the larger system. If a horizontal prototype is combined with a vertical prototype, both the level of functionality and the number of features are hence reduced resulting in an intersection that represents a cheap and fast to implement scenario. This product is shown in Figure 2 (overleaf).



Figure 2 – Scenario, obtained from [7]

Scenarios work best when used to simulate a predefined path or interaction to test the usability of a specific feature. And because they can be produced cheaply in a timely manner, they can be used frequently at the different development stages to get quick feedbacks from users [16].

5. Heuristic Evaluation

Jacob Nielsen introduced the concept of heuristic evaluation as an informal discount usability method that best suits those extremely time constrained and budget limited projects [16]. In heuristic evaluation, a *few* evaluators are given a *small* set of predefined principles "heuristics" and asked to independently analyze a user interface using these heuristics as a standard. Nielsen's heuristics are:

- 1. Visibility of system status: users are informed about what is going on.
- 2. Match between system and the real world: speak the users' language.
- 3. User control and freedom: Support undo and redo. Provide clear exits.
- 4. Consistency and standards: Follow platform and domain conventions.
- 5. Error prevention: eliminate error-prone conditions or check for them.
- 6. Recognition rather than recall: Minimize the user's memory load.
- 7. Flexibility and efficiency of use: Allow users to tailor frequent actions shortcuts.
- 8. Aesthetic and minimalist design: Do not provide irrelevant information
- 9. Help users recognize, diagnose, and recover from errors: Good error messages
- 10. Help and documentation: concise, easy to search and task-centered

Although it was empirically shown that 3 to 5 evaluators have proved to be sufficient as illustrated in Figure 3, increasing this number is strongly determined by the benefit brought to the evaluation process by recruiting one more evaluator.



Figure 3 - Usability Problems Found vs. Number of Evaluators, copied from [18]

Nielsen asserts that heuristic evaluation better be a group effort, simply because no one individual can inspect all the usability issues, but rather "different people find different usability problems" [19]. The evaluation session should take no longer than two hours. Then, results from all evaluators are reported in aggregate and then given back to the evaluators to give severity ratings for the problems identified. The final output of the evaluation session is a set of prioritized usability problems in the system under inspection. Optionally, by the end of the evaluation session, the evaluators can meet with the design team to discuss the usability issues and possible solutions.

Heuristic evaluation can be applied very early in the design cycle even before any implementation starts, particularly if prototypes are used for evaluation. It can also be applied during the actual implementation phase, especially in iterative development models like Agile, where usability inspection can be applied to ready-to-use interfaces.

Discount Usability in Agile Methods

Agile methods [20] were introduced and defended by software developers in response to a history full of frustration because of projects going beyond their time frame and budget limits and more importantly not meeting the customer's requirements [21]. Fortunately, software organizations adopting agile methods have been experiencing great success in meeting customers' requirements and producing useful pieces of software. However, in a competitive market, it is no longer sufficient to develop *useful* software the demand for *useful* and *usable* software has been dramatically increasing [22].

For agile methods in particular, it is important to focus on *discount* usability practices rather than general usability methods. This is mainly driven by the nature of agile development models that follow an iterative approach. Agile iterations are expected to respect a very strict time framework. Each iteration in Agile lasts from two to four weeks. During each iteration, a daily scrum meeting is to be held as an updating and information trading mechanism amongst the members of the Agile team. By the end of the iteration, a potentially shippable mini-release is expected to be available.

Constantine had one of the first attempts to relate Usage Centered Design (UCD) with Agile development [23]. A more serious call to build a bridge between usability and Agile methods was by Kane who explicitly encouraged the Agile community to "[find] a place for discount usability." His work was a reaction to the fact that he couldn't find a single article in Agile literature explicitly addressing usability, except for one article that briefly mentioned usability in its summary [7]. Kane asserts that there is a common ground between discount usability engineering and agile development, since they both attempt to give a special focus on the minimal crucial aspects of their respective domains. Thus, he suggested that Agile teams adopt the discount usability methods developed by Nielsen [24] and others. But he did not discuss how these methods can be practically employed in an Agile development model.

Possible Approach

A very recent report on "Adapting Usability Investigations for Agile User-Centered Design" has been published this year (2007) by Sy [22]. In this article, Sy talks about a remarkable experience with employing usability methods in an Agile environment. She explains in detail how her organization adopted an Agile approach while maintaining a usability centered design, which resulted in a highly usable product delivered on time. The main motivation behind this link between Agile and usability was the fact that the Agile team in the organization had a tested working version at almost any point of the development process. This made incremental usability evaluation of the product more attainable. To make this integration even more effective, the User Experience Team played the customer's role to ensure an unbiased evaluation of the usability testing in the



Figure 4 – Agile development process with usability testing, obtained from [22]

Although the organization has not adopted discount usability approaches only, their model can still be applied with the discount usability restrictions being taken into consideration. The following explanation is directly obtained from Sy but with a few modifications to make the model applicable with a discount usability approach.

Cycle Zero, as shown in the previous figure, is a quick high level requirement elicitation phase to share a common vision of the project among the development team members, the interaction designers and the customer. While developers try to spike on some technically questionable features, during this phase, interaction designers try to establish the initial design principles to enlighten the development process. If the product is completely novel, light and vivid scenarios can be developed to better investigate the users' needs. After Cycle Zero, the work flow splits into two tracks: developer track and interaction designer track. In Cycle 1, developers start coding the business logic layer. Meanwhile, interaction designers develop prototypes for user interfaces to be coded in the coming Cycles. They sometimes need to conduct rapid usability testing to validate their design decisions. When Cycle 2 starts, developers receive the interface prototypes developed in Cycle 1 and start coding accordingly. Simultaneously, interaction designers conduct usability testing on the currently available working system (from Cycle 1) and prepare new prototypes for the next Cycles. This pattern of designing at least a cycle ahead of developing continues until the product is released.

Methods

The case study discussed in this section originally incorporates a general usability approach rather than a discount one. Originally, the interaction designers in the organization used to conduct contextual queries about a set of features two cycles before the developers start coding that set. Conducting contextual queries is a time consuming usability investigation method that can in no way be considered discount. Considering a case where only discount usability methods are to be employed, the organization could have utilized several techniques such as think-aloud, scenarios, card sorting, taskcentered walkthroughs and heuristic evaluation. As mentioned earlier, during Cycle Zero, the interaction designers used quick scenarios to understand the users' needs if the product was novel. In later cycles, they could have used mock-up interface designs to conduct brief think-aloud sessions, especially if they were interested in knowing what consists a logical flow of processes to a typical user. They could have also utilized the card sorting technique to allow potential end users to group pieces of information and lists of commands in a way that might seem logical to them. This brief usability assurance guides the design process so that no excessive redesign is required later. The design issues addressed in each design cycle could be incorporated in the first development cycle to come.

After the end of each development cycle, a working version of the system featuring a subset of the customer's requirements is expected to be available. Therefore, the design team could have conducted a task-centered walkthrough session to assess the usability of the available features.

It would have been also beneficial if the team had allowed enough time before the release date to conduct a heuristic evaluation session and incorporate any suggested usability fixes within the time frame available.

Advantages

What makes adopting discount usability methods a widely acceptable and advantageous step is that they are easy to teach and easy to comprehend. For example, according to [18], it takes only a half-day seminar to explain the few guidelines that will reveal many usability problems in heuristic evaluation. These discount methods are also fast to conduct. A discount usability evaluation session does not take more than a few hours or effectively one working day in most cases, especially if quick task-centered walkthroughs or rapid think-aloud sessions are to be held.

Of high importance, discount methods are relatively cheap since no sophisticated laboratories or expensive equipment is required. Additionally, as opposed to the general usability approaches, most discount usability methods don't necessarily require usability experts or typical end users to perform the evaluation - recruiting evaluators is a very flexible process in these methods. Furthermore, the fact that some discount methods such as card sorting and heuristic evaluation do not necessarily require a working system makes it suitable for getting feedback early in the design process. Due to these advantageous characteristics, a very high benefit to cost ratio can be achieved when applying these methods. One of the studies conducted by [25] to justify the cost of conducting a discount heuristic evaluation using a conservative approach revealed a benefit to cost ratio of 48.

Disadvantages

Although discount usability methods have many pros, usability experts address some defects in these methods. One of the major problems is the oversimplification or distortion resulting from attempting to produce a simpler, easier or faster version from the original usability evaluation methods so that they can be described as discount. For instance, the usability heuristics list originally consisted of about a thousand entries [26] Nielsen; however, decided to come up with an optimized list of these principles in order to make it a discount usability method. This huge reduction made the new set of principles very broad and general [27]. This generality has a strong potential to create confusion among the different kinds of usability issues or mislead developers when trying to find specific solutions.

Moreover, although the fact that some discount methods do not necessarily need to recruit typical end users may be considered as a flexibility advantage, some research suggests that results not obtained from end users may lead to incorporating changes that are not realistically required [28]. For instance, [29] states that heuristic evaluation finds a "distressing" number of minor problems, some of which are only a matter of "taste," the thing that makes pointing out these problems a "false alarm." Therefore, changing the design to incorporate these false alarms may not only waste resources, but also, and more dangerously, degrade the usability of the system.

Conclusion

Until a very late decade, software organizations had perceived usability as an expensive and time consuming practice. Such understanding resulted in usability engineering being employed in a few multimillion dollar organizations only but totally avoided in smaller companies. Usability practitioners tried to treat this misconception by conveying that usability need not slow down your process or cost you a fortune to be effective. Nielsen; therefore, coined the term "discount usability" not as an alternative to existing usability approaches, but as a highly effectual and gratifying usability testing practice compared to doing nothing.

Discount usability generally consists of informal basic usability testing methodologies that can be conducted cheaply, rapidly and easily. These methods including card sorting, think-aloud, walkthroughs, scenarios and heuristic evaluation proved to be effective and rewarding. Mainly because they are relatively easy to learn, fast to conduct, cheap and flexible. Although it is an advantage, usability investigators need not be usability experts or domain experts to conduct a discount usability test.

Usability testing with discount methods can be applied early in the design phase using prototypes, or during the implementation phase using a working system. In an iterative design approach like Agile where requirement elicitation is a continuous process and customers' satisfaction is the ultimate goal, discount usability is particularly very handy. A variety of discount usability techniques can be utilized at any point of the project cycle since a working system or a mini-release is very likely to be available.

In general, discount usability may not provide for the best techniques to conduct indepth usability testing, especially with complex interfaces that require a comprehensive consideration of human factors or psychological patterns to uncover usability issues. Moreover, discount usability may not be suitable to test the usability of critical systems or to conduct a detailed analysis of users' expectations.

Discount usability is not expected to replace other usability engineering methods, but it can be an excellent motivation for organizations to penetrate the intimidation barrier and not avoid usability testing anymore. Also, the cost of usability can be more easily justified when discount methods are employed.

References

- Dumas, S. and Redish, C. (1999). A Practical Guide to Usability Testing. Revised Edition. Ch2.
- [2] Usability Makes a Come Back, Kheterpal, S. (2001). ClickZ, http://www.pcd-innovations.com/usability_evaluation.htm, accessed 6 Dec, 2007.
- [3] Greenberg, S. Interface Hall of Shame, http://pages.cpsc.ucalgary.ca/~saul/ wiki/uploads/CPSC681/RealPhoneAll.pdf, accessed 15 Oct, 2007.
- [4] Interface Hall of Shame (1997), http://homepage.mac.com/bradster/ iarchitect/phone.htm, accessed 11 Nov, 2007.
- [5] Redish, J. (2007). A bit history of usability testing and why it's not expensive any more, http://redish.net/writingfortheweb/index.php/2007/06/10/a-bit-of-history-ofusability-testing-and-why-its-not-so-expensive-any-more, accessed 11 Nov, 2007.
- [6] Nielsen, J. (1995). Applying Discount Usability Engineering. *IEEE Software*, vol. 12, no. 1, pp. 98-100.
- [7] Kane, D. Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet. *Proceedings of Agile Development Conference* (ADC'03), 2003.
- [8] Vredenburg, K., Mao, J.-Y., Smith, P. W., & Carey, T. A Survey of User-Centered Design Practice, Proc. CHI 2002, ACM Press (2002), 472-478
- [9] Think Aloud Protocol. (2007), http://en.wikipedia.org/wiki/Think_aloud_protocol, accessed 15 Nov, 2007.
- [10] Frøkjær, E., Hornbæk, K. (2005). Cooperative usability testing: complementing usability tests with user-supported interpretation sessions, *CHI '05 extended abstracts on Human factors in computing systems*, April 02-07, 2005, Portland, OR, USA.
- [11] Ericsson, K., & Simon, H. (1987). "Verbal reports on thinking", in In C. Faerch & G. Kasper (eds.): *Introspection in Second Language Research*. Clevedon, Avon: Multilingual Matters, 24–54
- [12] O'Malley, C., Draper, S. & Riley, M. Constructive Interaction: A Method for studying User-Computer-User Interaction. In Proc. of Interact, 1984, 1 -5.

- [13] Lewis, C., Polson, P., Wharton, C., Rieman, J. (1990) Testing a walkthrough methodology for theory-based designof walk-up-and-use interfaces. In CHI'90 Proceedings, Seattle, WA, ACM, NY.
- [14] Lewis & Reiman (1993) Task Centered User Interface Design: A Practical Introduction. University of Colorado, Boulder. Shareware book available from ftp.cs.colorado.edu/pub/cs/distribs/clewis/HCIDesign-Book/
- [15] Working Through Task-Centered System Design, Saul Greenberg
- [16] Nielsen, J., Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier, http://www.useit.com/papers/guerrilla_hci.html, accessed 18 Oct, 2007.
- [17] Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256.
- [18] Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, p25-62. John Wiley & Sons, New York, NY (1994).
- [19] Nielsen, J., Heuristic evaluation, http://www.useit.com/papers/heuristic/heuristic_evaluation.html, accessed September 30, 2007.
- [20] Schwaber, K., Beedle, M. (2002). Agile Software Development with Scrum. Upper Sadle River, NJ: Prentice Hall
- [21] Gibbs, W. W. Software's Chronic Crisis. Scientific American (Sept. 1994), pp. 86-95.
- [22] Sy, D. Adapting usability investigations for agile user-centered design. *Journal of Usability Studies*, Vol. 2, No. 3, May 2007.
- [23] Constantine, L. Methodological Agility, Software Development, Vol. 9, No.6, June 2001.
- [24] Nielsen, J. Usability engineering at a discount. In Salvendy, G., and Smith, M.J. (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, Elsevier Science Publishers, Amsterdam. 394-401, 1989.
- [25] Nielsen, J. (1993b). Iterative user-interface design, *IEEE Computer 1993*, 26 (11), 32–41.

- [26] Smith, S., Mosier, J. (1986) Guidelines for designing user interface software, htp://www.hcibib.org/sam/contents.html, accessed October 14, 2007.
- [27] Greenberg, S., Overview of Heuristic Evaluation, http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/CPSC681/Heuristic.ppt, accessed October 10, 2007.
- [28] Heuristic Evaluation, Usability.gov, http://www.usability.gov/methods/heuristiceval.html, accessed October 1, 2007.
- [29] Jeffries, R., Desurvire, H. (1992). Usability testing vs. Heuristic evaluation: Was there a contest? *SIGCHI Bulletin*, 24 (4), p39.