How to Conduct a Heuristic Evaluation

Heuristic evaluation (Nielsen and Molich, 1990; Nielsen 1994) is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").

In general, heuristic evaluation is difficult for a single individual to do because one person will never be able to find all the usability problems in an interface. Luckily, experience from many different projects has shown that different people find different usability problems. Therefore, it is possible to improve the effectiveness of the method significantly by involving multiple evaluators. Figure 1 shows an example from a case study of heuristic evaluation where 19 evaluators were used to find 16 usability problems in a voice response system allowing customers access to their bank accounts (Nielsen 1992). Each of the black squares in Figure 1 indicates the finding of one of the usability problems by one of the evaluators. The figure clearly shows that there is a substantial amount of nonoverlap between the sets of usability problems found by different evaluators. It is certainly true that some usability problems are so easy to find that they are found by almost everybody, but there are also some problems that are found by very few evaluators. Furthermore, one cannot just identify the best evaluator and rely solely on that person's findings. First, it is not necessarily true that the same person will be the best evaluator every time. Second, some of the hardest-to-find usability problems (represented by the leftmost columns in Figure 1) are found by evaluators who do not otherwise find many usability problems. Therefore, it is necessary to involve multiple evaluators in any heuristic evaluation (see below for a discussion of the best number of evaluators). My recommendation is normally to use three to five evaluators since one does not gain that much additional information by using larger numbers.

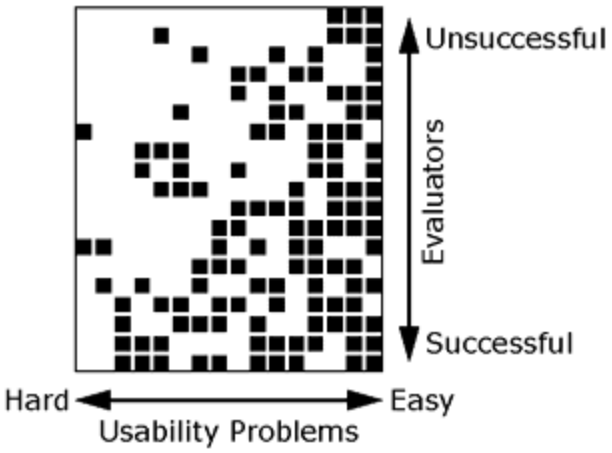

Figure 1

Illustration showing which evaluators found which usability problems in a heuristic evaluation of a banking system. Each row represents one of the 19 evaluators and each column represents one of the 16 usability problems. Each square shows whether the evaluator represented by the row found the usability problem represented by the column: The square is black if this is the case and white if the evaluator did not find the problem. The rows have been sorted in such a way that the most successful evaluators are at the bottom and the least successful are at the top. The columns have been sorted in such a way that the usability problems that are the easiest to find are to the right and the usability problems that are the most difficult to find are to the left.

Heuristic evaluation is performed by having each individual evaluator inspect the interface alone. Only after all evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated. This procedure is important in order to ensure independent and unbiased evaluations from each evaluator. The results of the evaluation can be recorded either as written reports from each evaluator or by having the evaluators verbalize their comments to an observer as they go through the interface. Written reports have the advantage of presenting a formal record of the evaluation, but require an additional effort by the evaluators and the need to be read and aggregated by an evaluation manager. Using an observer adds to the overhead of each evaluation session, but reduces the workload on the evaluators. Also, the results of the evaluation are available fairly soon after the last evaluation session since the observer only needs to understand and organize one set of personal notes, not a set of reports written by others. Furthermore, the observer can assist the evaluators in operating the interface in case of problems, such as an unstable prototype, and help if the evaluators have limited domain expertise and need to have certain aspects of the interface explained.

In a user test situation, the observer (normally called the "experimenter") has the responsibility of interpreting the user's actions in order to infer how these actions are related to the usability issues in the design of the interface. This makes it possible to conduct user testing even if the users do not know anything about user interface design. In contrast, the responsibility for analyzing the user interface is placed with the evaluator in a heuristic evaluation session, so a possible observer only needs to record the

evaluator's comments about the interface, but does not need to interpret the evaluator's actions.

Two further differences between heuristic evaluation sessions and traditional user testing are the willingness of the observer to answer questions from the evaluators during the session and the extent to which the evaluators can be provided with hints on using the interface. For traditional user testing, one normally wants to discover the mistakes users make when using the interface; the experimenters are therefore reluctant to provide more help than absolutely necessary. Also, users are requested to discover the answers to their questions by using the system rather than by having them answered by the experimenter. For the heuristic evaluation of a domain-specific application, it would be unreasonable to refuse to answer the evaluators' questions about the domain, especially if nondomain experts are serving as the evaluators. On the contrary, answering the evaluators' questions will enable them to better assess the usability of the user interface with respect to the characteristics of the domain. Similarly, when evaluators have problems using the interface, they can be given hints on how to proceed in order not to waste precious evaluation time struggling with the mechanics of the interface. It is important to note, however, that the evaluators should not be given help until they are clearly in trouble and have commented on the usability problem in question.

Typically, a heuristic evaluation session for an individual evaluator lasts one or two hours. Longer evaluation sessions might be necessary for larger or very complicated interfaces with a substantial number of dialogue elements, but it would be better to split up the evaluation into several smaller sessions, each concentrating on a part of the interface.

During the evaluation session, the evaluator goes through the interface several times and inspects the various dialogue elements and compares them with a list of recognized usability principles (the heuristics). These heuristics are general rules that seem to describe common properties of usable interfaces. In addition to the checklist of general heuristics to be considered for all dialogue elements, the evaluator obviously is also allowed to consider any additional usability principles or results that come to mind that may be relevant for any specific dialogue element. Furthermore, it is possible to develop category-specific heuristics that apply to a specific class of products as a supplement to the general heuristics. One way of building a supplementary list of category-specific heuristics is to perform competitive analysis and user testing of existing products in the given category and try to abstract principles to explain the usability problems that are found (Dykstra 1993).

In principle, the evaluators decide on their own how they want to proceed with evaluating the interface. A general recommendation would be that they go through the interface at least twice, however. The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.

Since the evaluators are not *using* the system as such (to perform a real task), it is possible to perform heuristic evaluation of user interfaces that exist on paper only and have not yet been implemented (Nielsen 1990). This makes heuristic evaluation suited for use early in the usability engineering lifecycle.

If the system is intended as a walk-up-and-use interface for the general population or if the evaluators are domain experts, it will be possible to let the evaluators use the system without further assistance. If the system is domain-dependent and the evaluators are fairly naive with respect to the domain of the system, it will be necessary to assist the evaluators to enable them to use the interface. One approach that has been applied successfully is to supply the evaluators with a typical usage scenario, listing the various steps a user would take to perform a sample set of realistic tasks. Such a scenario should be constructed on the basis of a task analysis of the actual users and their work in order to be as representative as possible of the eventual use of the system.

The output from using the heuristic evaluation method is a list of usability problems in the interface with references to those usability principles that were violated by the design in each case in the opinion of the evaluator. It is not sufficient for evaluators to simply say that they do not like something; they should explain why they do not like it with reference to the heuristics or to other usability results. The evaluators should try to be as specific as possible and should list each usability problem separately. For example, if there are three things wrong with a certain dialogue element, all three should be listed with reference to the various usability principles that explain why each particular aspect of the interface element is a usability problem. There are two main reasons to note each problem separately: First, there is a risk of repeating some problematic aspect of a dialogue element, even if it were to be completely replaced with a new design, unless one is aware of all its problems. Second, it may not be possible to fix all usability problems in an interface element or to replace it with a new design, but it could still be possible to fix some of the problems if they are all known.

Heuristic evaluation does not provide a systematic way to generate fixes to the usability problems or a way to assess the probable quality of any redesigns. However, because heuristic evaluation aims at explaining each observed usability problem with reference to established usability principles, it will often be fairly easy to generate a revised design according to the guidelines provided by the violated principle for good interactive systems. Also, many usability problems have fairly obvious fixes as soon as they have been identified.

For example, if the problem is that the user cannot copy information from one window to another, then the solution is obviously to include such a copy feature. Similarly, if the problem is the use of inconsistent typography in the form of upper/lower case formats and fonts, the solution is obviously to pick a single typographical format for the entire interface. Even for these simple examples, however, the designer has no information to help design the exact changes to the interface (e.g., how to enable the user to make the copies or on which of the two font formats to standardize).

One possibility for extending the heuristic evaluation method to provide some design advice is to conduct a debriefing session after the last evaluation session. The participants in the debriefing should include the evaluators, any observer used during the evaluation sessions, and representatives of the design team. The debriefing session would be conducted primarily in a brainstorming mode and would focus on discussions of possible redesigns to address the major usability problems and general problematic aspects of the design. A debriefing is also a good opportunity for discussing the positive aspects of the design, since heuristic evaluation does not otherwise address this important issue.

Heuristic evaluation is explicitly intended as a "discount usability engineering" method. Independent research (Jeffries et al. 1991) has indeed confirmed that heuristic evaluation is a very efficient usability engineering method. One of my case study found a benefit-cost ratio for a heuristic evaluation project of 48: The cost of using the method was about $10,500 and the expected benefits were about $500,000 (Nielsen 1994). As a discount usability engineering method, heuristic evaluation is not guaranteed to provide "perfect" results or to find every last usability problem in an interface.

Determining the Number of Evaluators

In principle, individual evaluators can perform a heuristic evaluation of a user interface on their own, but the experience from several projects indicates that fairly poor results are achieved when relying on single evaluators. Averaged over six of my projects, single evaluators found only 35 percent of the usability problems in the interfaces. However, since different evaluators tend to find different problems, it is possible to achieve substantially better performance by aggregating the evaluations from several evaluators. Figure 2 shows the proportion of usability problems found as more and more evaluators are added. The figure clearly shows that there is a nice payoff from using more than one evaluator. It would seem reasonable to recommend the use of about five evaluators, but certainly at least three. The exact number of evaluators to use would depend on a cost-benefit analysis. More evaluators should obviously be used in cases where usability is critical or when large payoffs can be expected due to extensive or mission-critical use of a system.

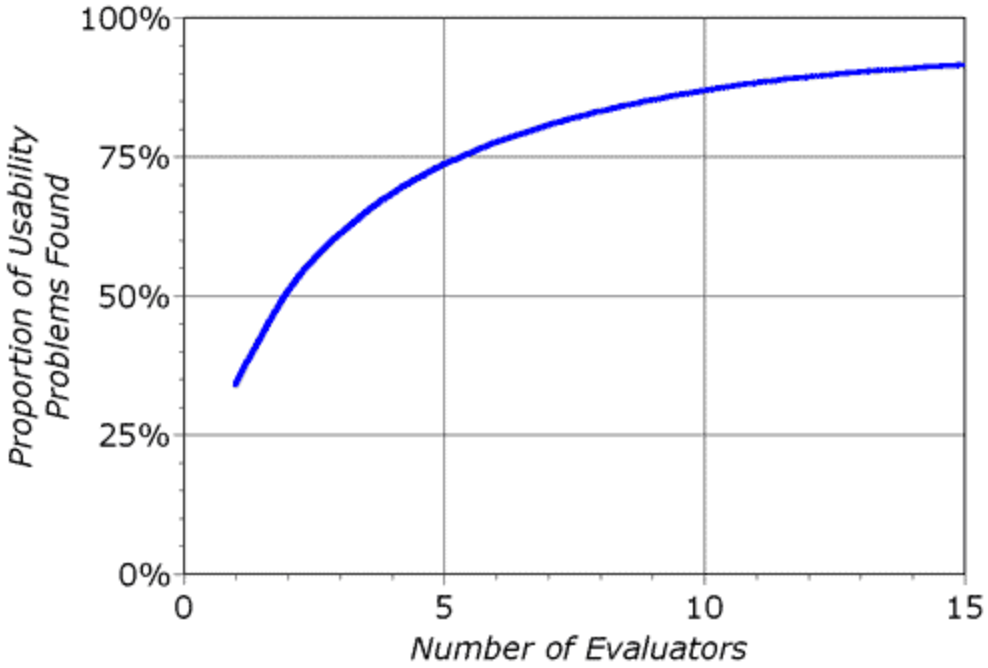

Figure 2
Curve showing the proportion of usability problems in an interface found by heuristic evaluation using various numbers of evaluators. The curve represents the average of six case studies of heuristic evaluation.

Nielsen and Landauer (1993) present such a model based on the following prediction formula for the number of usability problems found in a heuristic evaluation:

$$\text{ProblemsFound}(i) = N(1 - (1-l)^i)$$

where ProblemsFound(*i*) indicates the number of different usability problems found by aggregating reports from *i* independent evaluators, N indicates the total number of usability problems in the interface, and l indicates the proportion of all usability problems found by a single evaluator. In six case studies (Nielsen and Landauer 1993), the values of l ranged from 19 percent to 51 percent with a mean of 34 percent. The values of N ranged from 16 to 50 with a mean of 33. Using this formula results in curves very much like that shown in Figure 2, though the exact shape of the curve will vary with the values of the parameters N and l, which again will vary with the characteristics of the project.

In order to determine the optimal number of evaluators, one needs a cost-benefit model of heuristic evaluation. The first element in such a model is an accounting for the cost of using the method, considering both fixed and variable costs. Fixed costs are those that need to be paid no matter how many evaluators are used; these include time to plan the evaluation, get the materials ready, and write up the report or otherwise communicate the results. Variable costs are those additional costs that accrue each time one additional evaluator is used; they include the loaded salary of that evaluator as well as the cost of analyzing the evaluator's report and the cost of any computer or other resources used

during the evaluation session. Based on published values from several projects the fixed cost of a heuristic evaluation is estimated to be between $3,700 and $4,800 and the variable cost of each evaluator is estimated to be between $410 and $900.

The actual fixed and variable costs will obviously vary from project to project and will depend on each company's cost structure and on the complexity of the interface being evaluated. For illustration, consider a sample project with fixed costs for heuristic evaluation of $4,000 and variable costs of $600 per evaluator. In this project, the cost of using heuristic evaluation with `i` evaluators is thus `$(4,000 + 600i)`.

The benefits from heuristic evaluation are mainly due to the finding of usability problems, though some continuing education benefits may be realized to the extent that the evaluators increase their understanding of usability by comparing their own evaluation reports with those of other evaluators. For this sample project, assume that it is worth $15,000 to find each usability problem, using a value derived by Nielsen and Landauer (1993) from several published studies. For real projects, one would obviously need to estimate the value of finding usability problems based on the expected user population. For software to be used in-house, this value can be estimated based on the expected increase in user productivity; for software to be sold on the open market, it can be estimated based on the expected increase in sales due to higher user satisfaction or better review ratings. Note that real value only derives from those usability problems that are in fact fixed before the software ships. Since it is impossible to fix all usability problems, the value of each problem found is only some proportion of the value of a fixed problem.

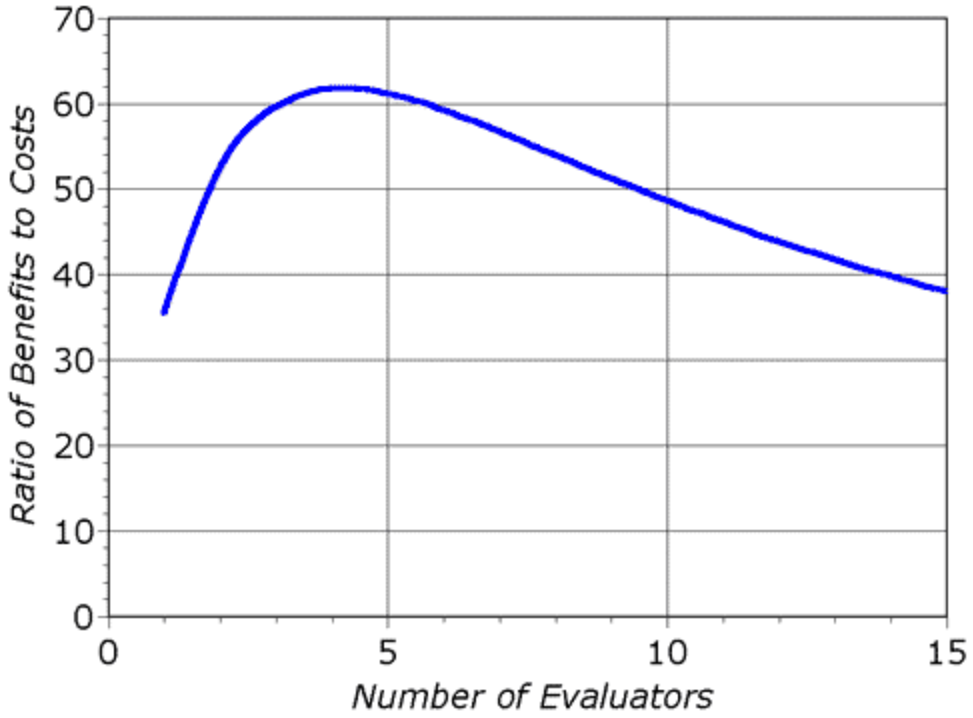

Figure 3

Curve showing how many times the benefits are greater than the costs for heuristic evaluation of a sample project using the assumptions discussed in the text. The optimal number of evaluators in this example is four, with benefits that are 62 times greater than the costs.

Figure 3 shows the varying ratio of the benefits to the costs for various numbers of evaluators in the sample project. The curve shows that the optimal number of evaluators in this example is four, confirming the general observation that heuristic evaluation seems to work best with three to five evaluators. In the example, a heuristic evaluation with four evaluators would cost $6,400 and would find usability problems worth $395,000.

References

- Dykstra, D. J. 1993. *A Comparison of Heuristic Evaluation and Usability Testing: The Efficacy of a Domain-Specific Heuristic Checklist*. Ph.D. diss., Department of Industrial Engineering, Texas A&M University, College Station, TX.
- Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. 1991. User interface evaluation in the real world: A comparison of four techniques. *Proceedings ACM CHI'91 Conference* (New Orleans, LA, April 28-May 2), 119-124.
- Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* **33**, 3 (March), 338-348.
- Nielsen, J. 1990. Paper versus computer implementations as mockup scenarios for heuristic evaluation. *Proc. IFIP INTERACT90 Third Intl. Conf. Human-Computer Interaction* (Cambridge, U.K., August 27-31), 315-320.
- Nielsen, J., and Landauer, T. K. 1993. A mathematical model of the finding of usability problems. *Proceedings ACM/IFIP INTERCHI'93 Conference* (Amsterdam, The Netherlands, April 24-29), 206-213.
- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.
- Nielsen, J. 1992. Finding usability problems through heuristic evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7), 373-380.
- Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, NY.

How to Conduct a Heuristic Evaluation

Heuristic evaluation (Nielsen and Molich, 1990; Nielsen 1994) is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a

small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").

In general, heuristic evaluation is difficult for a single individual to do because one person will never be able to find all the usability problems in an interface. Luckily, experience from many different projects has shown that different people find different usability problems. Therefore, it is possible to improve the effectiveness of the method significantly by involving multiple evaluators. Figure 1 shows an example from a case study of heuristic evaluation where 19 evaluators were used to find 16 usability problems in a voice response system allowing customers access to their bank accounts (Nielsen 1992). Each of the black squares in Figure 1 indicates the finding of one of the usability problems by one of the evaluators. The figure clearly shows that there is a substantial amount of nonoverlap between the sets of usability problems found by different evaluators. It is certainly true that some usability problems are so easy to find that they are found by almost everybody, but there are also some problems that are found by very few evaluators. Furthermore, one cannot just identify the best evaluator and rely solely on that person's findings. First, it is not necessarily true that the same person will be the best evaluator every time. Second, some of the hardest-to-find usability problems (represented by the leftmost columns in Figure 1) are found by evaluators who do not otherwise find many usability problems. Therefore, it is necessary to involve multiple evaluators in any heuristic evaluation (see below for a discussion of the best number of evaluators). My recommendation is normally to use three to five evaluators since one does not gain that much additional information by using larger numbers.

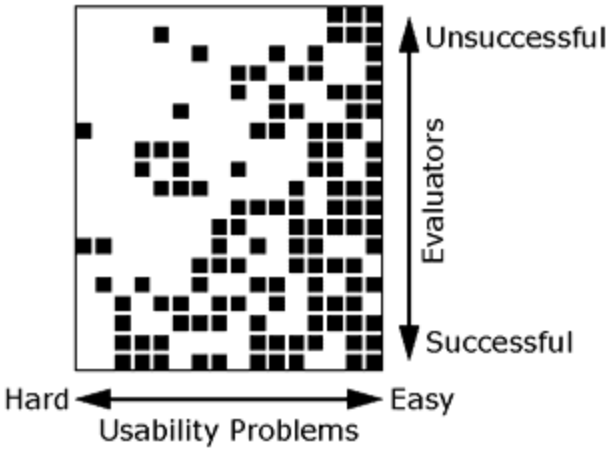

Figure 1

Illustration showing which evaluators found which usability problems in a heuristic evaluation of a banking system. Each row represents one of the 19 evaluators and each column represents one of the 16 usability problems. Each square shows whether the evaluator represented by the row found the usability problem represented by the column: The square is black if this is the case and white if the evaluator did not find the problem. The rows have been sorted in such a way that the most successful evaluators are at the bottom and the least successful are at the top. The columns have been sorted in such a way that the usability problems that are the easiest to find are to the right and the usability problems that are the most difficult to find are to the left.

Heuristic evaluation is performed by having each individual evaluator inspect the interface alone. Only after all evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated. This procedure is important in order to ensure independent and unbiased evaluations from each evaluator. The results of the evaluation can be recorded either as written reports from each evaluator or by having the evaluators verbalize their comments to an observer as they go through the interface. Written reports have the advantage of presenting a formal record of the evaluation, but require an additional effort by the evaluators and the need to be read and aggregated by an evaluation manager. Using an observer adds to the overhead of each evaluation session, but reduces the workload on the evaluators. Also, the results of the evaluation are available fairly soon after the last evaluation session since the observer only needs to understand and organize one set of personal notes, not a set of reports written by others. Furthermore, the observer can assist the evaluators in operating the interface in case of problems, such as an unstable prototype, and help if the evaluators have limited domain expertise and need to have certain aspects of the interface explained.

In a user test situation, the observer (normally called the "experimenter") has the responsibility of interpreting the user's actions in order to infer how these actions are related to the usability issues in the design of the interface. This makes it possible to conduct user testing even if the users do not know anything about user interface design. In contrast, the responsibility for analyzing the user interface is placed with the evaluator in a heuristic evaluation session, so a possible observer only needs to record the evaluator's comments about the interface, but does not need to interpret the evaluator's actions.

Two further differences between heuristic evaluation sessions and traditional user testing are the willingness of the observer to answer questions from the evaluators during the session and the extent to which the evaluators can be provided with hints on using the interface. For traditional user testing, one normally wants to discover the mistakes users make when using the interface; the experimenters are therefore reluctant to provide more help than absolutely necessary. Also, users are requested to discover the answers to their questions by using the system rather than by having them answered by the experimenter. For the heuristic evaluation of a domain-specific application, it would be unreasonable to refuse to answer the evaluators' questions about the domain, especially if nondomain experts are serving as the evaluators. On the contrary, answering the evaluators' questions will enable them to better assess the usability of the user interface with respect to the characteristics of the domain. Similarly, when evaluators have problems using the interface, they can be given hints on how to proceed in order not to waste precious evaluation time struggling with the mechanics of the interface. It is important to note, however, that the evaluators should not be given help until they are clearly in trouble and have commented on the usability problem in question.

Typically, a heuristic evaluation session for an individual evaluator lasts one or two hours. Longer evaluation sessions might be necessary for larger or very complicated interfaces with a substantial number of dialogue elements, but it would be better to split

up the evaluation into several smaller sessions, each concentrating on a part of the interface.

During the evaluation session, the evaluator goes through the interface several times and inspects the various dialogue elements and compares them with a list of recognized usability principles (the heuristics). These heuristics are general rules that seem to describe common properties of usable interfaces. In addition to the checklist of general heuristics to be considered for all dialogue elements, the evaluator obviously is also allowed to consider any additional usability principles or results that come to mind that may be relevant for any specific dialogue element. Furthermore, it is possible to develop category-specific heuristics that apply to a specific class of products as a supplement to the general heuristics. One way of building a supplementary list of category-specific heuristics is to perform competitive analysis and user testing of existing products in the given category and try to abstract principles to explain the usability problems that are found (Dykstra 1993).

In principle, the evaluators decide on their own how they want to proceed with evaluating the interface. A general recommendation would be that they go through the interface at least twice, however. The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.

Since the evaluators are not *using* the system as such (to perform a real task), it is possible to perform heuristic evaluation of user interfaces that exist on paper only and have not yet been implemented (Nielsen 1990). This makes heuristic evaluation suited for use early in the usability engineering lifecycle.

If the system is intended as a walk-up-and-use interface for the general population or if the evaluators are domain experts, it will be possible to let the evaluators use the system without further assistance. If the system is domain-dependent and the evaluators are fairly naive with respect to the domain of the system, it will be necessary to assist the evaluators to enable them to use the interface. One approach that has been applied successfully is to supply the evaluators with a typical usage scenario, listing the various steps a user would take to perform a sample set of realistic tasks. Such a scenario should be constructed on the basis of a task analysis of the actual users and their work in order to be as representative as possible of the eventual use of the system.

The output from using the heuristic evaluation method is a list of usability problems in the interface with references to those usability principles that were violated by the design in each case in the opinion of the evaluator. It is not sufficient for evaluators to simply say that they do not like something; they should explain why they do not like it with reference to the heuristics or to other usability results. The evaluators should try to be as specific as possible and should list each usability problem separately. For example, if there are three things wrong with a certain dialogue element, all three should be listed with reference to the various usability principles that explain why each particular aspect

of the interface element is a usability problem. There are two main reasons to note each problem separately: First, there is a risk of repeating some problematic aspect of a dialogue element, even if it were to be completely replaced with a new design, unless one is aware of all its problems. Second, it may not be possible to fix all usability problems in an interface element or to replace it with a new design, but it could still be possible to fix some of the problems if they are all known.

Heuristic evaluation does not provide a systematic way to generate fixes to the usability problems or a way to assess the probable quality of any redesigns. However, because heuristic evaluation aims at explaining each observed usability problem with reference to established usability principles, it will often be fairly easy to generate a revised design according to the guidelines provided by the violated principle for good interactive systems. Also, many usability problems have fairly obvious fixes as soon as they have been identified.

For example, if the problem is that the user cannot copy information from one window to another, then the solution is obviously to include such a copy feature. Similarly, if the problem is the use of inconsistent typography in the form of upper/lower case formats and fonts, the solution is obviously to pick a single typographical format for the entire interface. Even for these simple examples, however, the designer has no information to help design the exact changes to the interface (e.g., how to enable the user to make the copies or on which of the two font formats to standardize).

One possibility for extending the heuristic evaluation method to provide some design advice is to conduct a debriefing session after the last evaluation session. The participants in the debriefing should include the evaluators, any observer used during the evaluation sessions, and representatives of the design team. The debriefing session would be conducted primarily in a brainstorming mode and would focus on discussions of possible redesigns to address the major usability problems and general problematic aspects of the design. A debriefing is also a good opportunity for discussing the positive aspects of the design, since heuristic evaluation does not otherwise address this important issue.

Heuristic evaluation is explicitly intended as a "discount usability engineering" method. Independent research (Jeffries et al. 1991) has indeed confirmed that heuristic evaluation is a very efficient usability engineering method. One of my case study found a benefit-cost ratio for a heuristic evaluation project of 48: The cost of using the method was about $10,500 and the expected benefits were about $500,000 (Nielsen 1994). As a discount usability engineering method, heuristic evaluation is not guaranteed to provide "perfect" results or to find every last usability problem in an interface.

Determining the Number of Evaluators

In principle, individual evaluators can perform a heuristic evaluation of a user interface on their own, but the experience from several projects indicates that fairly poor results are achieved when relying on single evaluators. Averaged over six of my projects, single evaluators found only 35 percent of the usability problems in the interfaces. However,

since different evaluators tend to find different problems, it is possible to achieve substantially better performance by aggregating the evaluations from several evaluators. Figure 2 shows the proportion of usability problems found as more and more evaluators are added. The figure clearly shows that there is a nice payoff from using more than one evaluator. It would seem reasonable to recommend the use of about five evaluators, but certainly at least three. The exact number of evaluators to use would depend on a cost-benefit analysis. More evaluators should obviously be used in cases where usability is critical or when large payoffs can be expected due to extensive or mission-critical use of a system.

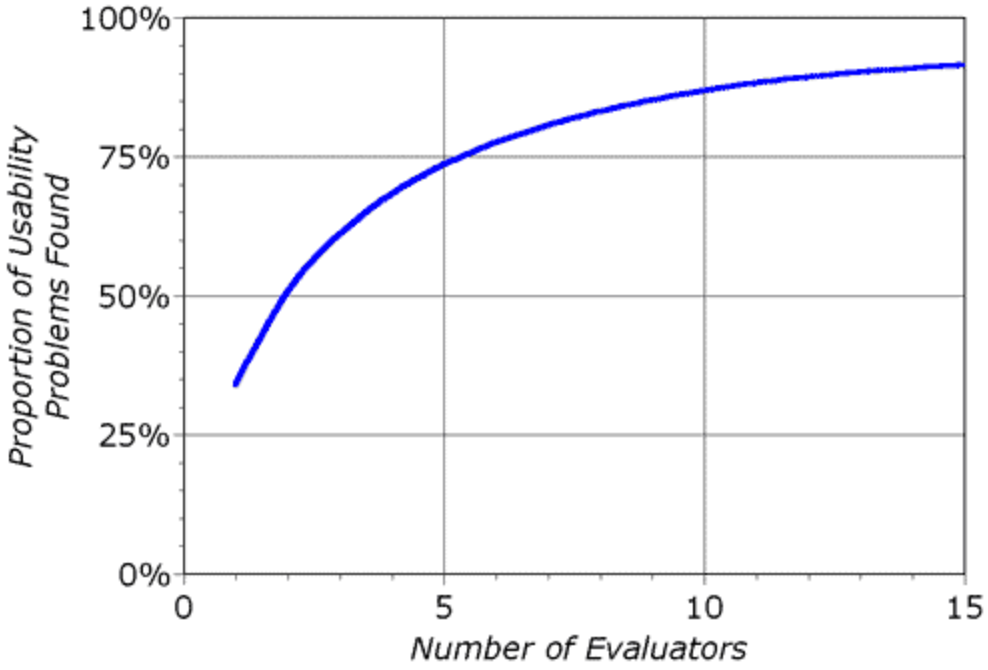

Figure 2
Curve showing the proportion of usability problems in an interface found by heuristic evaluation using various numbers of evaluators. The curve represents the average of six case studies of heuristic evaluation.

Nielsen and Landauer (1993) present such a model based on the following prediction formula for the number of usability problems found in a heuristic evaluation:

$$\mathtt{ProblemsFound}(i) = \mathtt{N}(1 - (1-\mathtt{l})^{i})$$

where `ProblemsFound(`*i*`)` indicates the number of different usability problems found by aggregating reports from *i* independent evaluators, `N` indicates the total number of usability problems in the interface, and l indicates the proportion of all usability problems found by a single evaluator. In six case studies (Nielsen and Landauer 1993), the values of `l` ranged from 19 percent to 51 percent with a mean of 34 percent. The values of `N` ranged from 16 to 50 with a mean of 33. Using this formula results in curves very much like that shown in Figure 2, though the exact shape of the curve will vary with the values of the parameters `N` and `l`, which again will vary with the characteristics of the project.

In order to determine the optimal number of evaluators, one needs a cost-benefit model of heuristic evaluation. The first element in such a model is an accounting for the cost of using the method, considering both fixed and variable costs. Fixed costs are those that need to be paid no matter how many evaluators are used; these include time to plan the evaluation, get the materials ready, and write up the report or otherwise communicate the results. Variable costs are those additional costs that accrue each time one additional evaluator is used; they include the loaded salary of that evaluator as well as the cost of analyzing the evaluator's report and the cost of any computer or other resources used during the evaluation session. Based on published values from several projects the fixed cost of a heuristic evaluation is estimated to be between $3,700 and $4,800 and the variable cost of each evaluator is estimated to be between $410 and $900.

The actual fixed and variable costs will obviously vary from project to project and will depend on each company's cost structure and on the complexity of the interface being evaluated. For illustration, consider a sample project with fixed costs for heuristic evaluation of $4,000 and variable costs of $600 per evaluator. In this project, the cost of using heuristic evaluation with i evaluators is thus $\$(4{,}000 + 600i)$.

The benefits from heuristic evaluation are mainly due to the finding of usability problems, though some continuing education benefits may be realized to the extent that the evaluators increase their understanding of usability by comparing their own evaluation reports with those of other evaluators. For this sample project, assume that it is worth $15,000 to find each usability problem, using a value derived by Nielsen and Landauer (1993) from several published studies. For real projects, one would obviously need to estimate the value of finding usability problems based on the expected user population. For software to be used in-house, this value can be estimated based on the expected increase in user productivity; for software to be sold on the open market, it can be estimated based on the expected increase in sales due to higher user satisfaction or better review ratings. Note that real value only derives from those usability problems that are in fact fixed before the software ships. Since it is impossible to fix all usability problems, the value of each problem found is only some proportion of the value of a fixed problem.

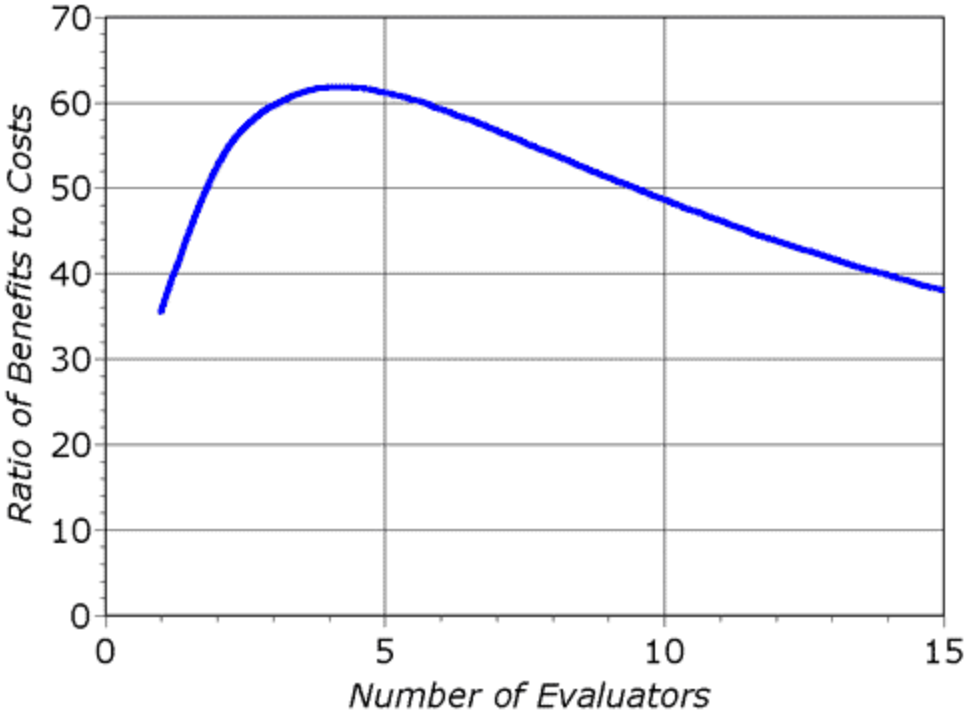

Figure 3

Curve showing how many times the benefits are greater than the costs for heuristic evaluation of a sample project using the assumptions discussed in the text. The optimal number of evaluators in this example is four, with benefits that are 62 times greater than the costs.

Figure 3 shows the varying ratio of the benefits to the costs for various numbers of evaluators in the sample project. The curve shows that the optimal number of evaluators in this example is four, confirming the general observation that heuristic evaluation seems to work best with three to five evaluators. In the example, a heuristic evaluation with four evaluators would cost $6,400 and would find usability problems worth $395,000.

References

- Dykstra, D. J. 1993. *A Comparison of Heuristic Evaluation and Usability Testing: The Efficacy of a Domain-Specific Heuristic Checklist*. Ph.D. diss., Department of Industrial Engineering, Texas A&M University, College Station, TX.
- Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. 1991. User interface evaluation in the real world: A comparison of four techniques. *Proceedings ACM CHI'91 Conference* (New Orleans, LA, April 28-May 2), 119-124.
- Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* **33**, 3 (March), 338-348.
- Nielsen, J. 1990. Paper versus computer implementations as mockup scenarios for heuristic evaluation. *Proc. IFIP INTERACT90 Third Intl. Conf. Human-Computer Interaction* (Cambridge, U.K., August 27-31), 315-320.

- Nielsen, J., and Landauer, T. K. 1993. A mathematical model of the finding of usability problems. *Proceedings ACM/IFIP INTERCHI'93 Conference* (Amsterdam, The Netherlands, April 24-29), 206-213.
- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.
- Nielsen, J. 1992. Finding usability problems through heuristic evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7), 373-380.
- Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, NY.

Ten Usability Heuristics

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Recognition rather than recall

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should

be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

I originally developed the heuristics for heuristic evaluation in collaboration with Rolf Molich in 1990 [Molich and Nielsen 1990; Nielsen and Molich 1990]. I since refined the heuristics based on a factor analysis of 249 usability problems [Nielsen 1994a] to derive a set of heuristics with maximum explanatory power, resulting in this revised set of heuristics [Nielsen 1994b].

See Also:

- Bruce "Tog" Tognazzini's list of basic principles for interface design. The list is slightly too long for heuristic evaluation but serves as a useful checklist.

References

- Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* **33**, 3 (March), 338-348.
- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.
- Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. *Proc. ACM CHI'94 Conf.* (Boston, MA, April 24-28), 152-158.
- Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

Characteristics of Usability Problems Found by Heuristic Evaluation

Heuristic evaluation is a good method for finding both major and minor problems in a user interface. As one might have expected, major problems are slightly easier to find than minor problems, with the probability for finding a given **major** usability problem at 42 percent on the average for single evaluators in six case studies (Nielsen 1992). The corresponding probability for finding a given **minor** problem was only 32 percent.

Even though major problems are easier to find, this does not mean that the evaluators concentrate exclusively on the major problems. In case studies of six user interfaces (Nielsen 1992), heuristic evaluation identified a total of 59 major usability problems and 152 minor usability problems. Thus, it is apparent that the lists of usability problems found by heuristic evaluation will tend to be dominated by minor problems, which is one reason severity ratings form a useful supplement to the method. Even though major

usability problems are by definition the most important ones to find and to fix, minor usability problems are still relevant. Many such minor problems seem to be easier to find by heuristic evaluation than by other methods. One example of such a minor problem found by heuristic evaluation was the use of inconsistent typography in two parts of a user interface. The same information would sometimes be shown in a serif font (like this one) and sometimes in a sans serif font (like this one), thus slowing users down a little bit as they have to expend additional effort on matching the two pieces of information. This type of minor usability problem could not be observed in a user test unless an extremely careful analysis were performed on the basis of a large number of videotaped or logged interactions, since the slowdown is very small and would not stop users from completing their tasks.

Usability problems can be located in a dialogue in four different ways: at a single location in the interface, at two or more locations that have to be compared to find the problem, as a problem with the overall structure of the interface, and finally as something that ought to be included in the interface but is currently missing. An analysis of 211 usability problems (Nielsen 1992) found that the difference between the four location categories was small and not statistically significant. In other words, evaluators were approximately equally good at finding all four kinds of usability problems. However, the interaction effect between location category and interface implementation was significant and had a very large effect. Problems in the category "something missing" were slightly easier to find than other problems in running systems, but much harder to find than other problems in paper prototypes. This finding corresponds to an earlier, qualitative, analysis of the usability problems that were harder to find in a paper implementation than in a running system (Nielsen 1990). Because of this difference, one should look harder for missing dialogue elements when evaluating paper mock-ups.

A likely explanation of this phenomenon is that evaluators using a running system may tend to get stuck when needing a missing interface element (and thus notice it), whereas evaluators of a paper "implementation" just turn to the next page and focus on the interface elements found there.

Alternating Heuristic Evaluation and User Testing

Even though heuristic evaluation finds many usability problems that are not found by user testing, it is also the case that it may miss some problems that can be found by user testing. Evaluators are probably especially likely to overlook usability problems if the system is highly domain-dependent and they have little domain expertise. In my case studies, including some that were so domain-specific that they would have been virtually impossible to find without user testing.

Since heuristic evaluation and user testing each finds usability problems overlooked by the other method, it is recommended that both methods be used. Because there is no reason to spend resources on evaluating an interface with many known usability problems only to have many of them come up again, it is normally best to use iterative design

between uses of the two evaluation methods. Typically, one would first perform a heuristic evaluation to clean up the interface and remove as many "obvious" usability problems as possible. After a redesign of the interface, it would be subjected to user testing both to check the outcome of the iterative design step and to find remaining usability problems that were not picked up by the heuristic evaluation.

There are two major reasons for alternating between heuristic evaluation and user testing as suggested here. First, a heuristic evaluation pass can eliminate a number of usability problems without the need to "waste users," who sometimes can be difficult to find and schedule in large numbers. Second, these two categories of usability assessment methods have been shown to find fairly distinct sets of usability problems; therefore, they supplement each other rather than lead to repetitive findings (Desurvire et al. 1992; Jeffries et al. 1991; Karat et al. 1992).

As another example, consider a video telephone system for interconnecting offices (Cool et al. 1992). Such a system has the potential for changing the way people work and interact, but these changes will become clear only after an extended usage period. Also, as with many computer-supported cooperative work applications, video telephones require a critical mass of users for the test to be realistic: If most of the people you want to call do not have a video connection, you will not rely on the system. Thus, on the one hand field testing is necessary to learn about changes in the users' long-term behavior, but on the other hand such studies will be very expensive. Therefore, one will want to supplement them with heuristic evaluation and laboratory-based user testing so that the larger field population does not have to suffer from glaring usability problems that could have been found much more cheaply. Iterative design of such a system will be a combination of a few, longer-lasting "outer iterations" with field testing and a larger number of more rapid "inner iterations" that are used to polish the interface before it is released to the field users.

References

- Cool, C., Fish, R. S., Kraut, R. E., and Lowery, C. M. 1992. Iterative design of video communication systems. *Proc. ACM CSCW'92 Conf. Computer-Supported Cooperative Work* (Toronto, Canada, November 1-4): 25-32.
- Desurvire, H. W., Kondziela, J. M., and Atwood, M. E. 1992. What is gained and lost when using evaluation methods other than empirical testing. In *People and Computers VII*, edited by Monk, A., Diaper, D., and Harrison, M. D., 89-102. Cambridge: Cambridge University Press. A shorter version of this paper is available in the *Digest of Short Talks presented at CHI'92* (Monterey, CA, May 7): 125-126.
- Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. 1991. User interface evaluation in the real world: A comparison of four techniques. *Proceedings ACM CHI'91 Conference* (New Orleans, LA, April 28-May 2): 119-124.
- Karat, C., Campbell, R. L., and Fiegel, T. 1992. Comparison of empirical testing and walkthrough methods in user interface evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7): 397-404.

- Nielsen, J. 1990. Paper versus computer implementations as mockup scenarios for heuristic evaluation. *Proc. IFIP INTERACT'90 Third Intl. Conf. Human-Computer Interaction* (Cambridge, U.K., August 27-31): 315-320.
- Nielsen, J. 1992. Finding usability problems through heuristic evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, May 3-7): 373-380.

Severity Ratings for Usability Problems

Severity ratings can be used to allocate the most resources to fix the most serious problems and can also provide a rough estimate of the need for additional usability efforts. If the severity ratings indicate that several disastrous usability problems remain in an interface, it will probably be unadvisable to release it. But one might decide to go ahead with the release of a system with several usability problems if they are all judged as being cosmetic in nature.

The severity of a usability problem is a combination of three factors:

- The **frequency** with which the problem occurs: Is it common or rare?
- The **impact** of the problem if it occurs: Will it be easy or difficult for the users to overcome?
- The **persistence** of the problem: Is it a one-time problem that users can overcome once they know about it or will users repeatedly be bothered by the problem?

Finally, of course, one needs to assess the **market impact** of the problem since certain usability problems can have a devastating effect on the popularity of a product, even if they are "objectively" quite easy to overcome. Even though severity has several components, it is common to combine all aspects of severity in a single severity rating as an overall assessment of each usability problem in order to facilitate prioritizing and decision-making.

The following 0 to 4 rating scale can be used to rate the severity of usability problems:

0 = I don't agree that this is a usability problem at all
1 = Cosmetic problem only: need not be fixed unless extra time is available on project
2 = Minor usability problem: fixing this should be given low priority
3 = Major usability problem: important to fix, so should be given high priority
4 = Usability catastrophe: imperative to fix this before product can be released

Severity Ratings in Heuristic Evaluation

It is difficult to get good severity estimates from the evaluators during a heuristic evaluation session when they are more focused on finding new usability problems. Also, each evaluator will only find a small number of the usability problems, so a set of severity ratings of only the problems found by that evaluator will be incomplete. Instead, severity ratings can be collected by sending a questionnaire to the evaluators after the actual evaluation sessions, listing the complete set of usability problems that have been discovered, and asking them to rate the severity of each problem. Since each evaluator has only identified a subset of the problems included in the list, the problems need to be described in reasonable depth, possibly using screendumps as illustrations. The descriptions can be synthesized by the evaluation observer from the aggregate of comments made by those evaluators who had found each problem (or, if written evaluation reports are used, the descriptions can be synthesized from the descriptions in the reports). These descriptions allow the evaluators to assess the various problems fairly easily even if they have not found them in their own evaluation session. Typically, evaluators need only spend about 30 minutes to provide their severity ratings. It is important to note that each evaluator should provide individual severity ratings independently of the other evaluators.

Often, the evaluators will not have access to the actual system while they are considering the severity of the various usability problems. It is possible that the evaluators can gain additional insights by revisiting parts of the running interface rather than relying on their memory and the written problem descriptions. At the same time, there is no doubt that the evaluators will be slower at arriving at the severity ratings if they are given the option of interacting further with the system. Also, scheduling problems will sometimes make it difficult to provide everybody with computer access at convenient times if special computer resources are needed to run a prototype system or if software distribution is limited due to confidentiality considerations.

My experience indicates that severity ratings from a single evaluator are too unreliable to be trusted. As more evaluators are asked to judge the severity of usability problems, the quality of the mean severity rating increases rapidly, and using the **mean of a set of ratings from three evaluators** is satisfactory for many practical purposes.

Technology Transfer of Heuristic Evaluation and Usability Inspection

by Jakob Nielsen

This paper was originally presented as a keynote at the IFIP **INTERACT'95** International Conference on Human-Computer Interaction (Lillehammer, Norway, June 27, 1995).

Abstract

Participants in a course on usability inspection methods were surveyed 7-8 months after the course to find out what methods they were in fact using, and why they used or did not use the methods they had been taught. The major factor in method usage was the quality of the usability information gained from the method, with a very strong correlation between the rated benefit of using a method and the number of times the method had been used. Even though the respondents came from companies with above-average usability budgets (7% of development budgets were devoted to usability), the cost of using the methods was also a very strong factor in determining use. Other observations were that technology transfer was most successful when methods were taught at the time when people had a specific need for them in their project, and that methods need to have active evangelists to succeed.

The Need for More Usable Usability

User interface professionals ought to take their own medicine some more. How often have we heard UI folks complain that "we get no respect" (from development managers)? At the same time, we have nothing but scorn for any programmer who has the attitude that if users have problems with his or her program then it must be the users' fault.

If we consider usability engineering as a system, a design, or a set of interfaces with which development managers have to interact, then it obviously becomes the usability professionals' responsibility to design that system to maximize its communication with its users. My claim is that any problems in getting usability results used more in development are more due to lack of usability of the usability methods and results than they are caused by evil development managers who deliberately want to torment their users.

In order to get usability methods used more in real development projects, we must make the usability methods easier to use and more attractive. One way of doing so is to consider the way current usability methods are being used and what causes some methods to be used and others to remain "a good idea which we might try on the next project." As an example of such studies I will report on a study of what causes usability inspection methods to be used.

Usability Inspection Methods

Usability inspection (Nielsen and Mack, 1994) is the generic name for a set of methods based on having evaluators inspect or examine usability-related aspects of a user interface. Some evaluators can be usability specialists, but they can also be software development consultants with special expertise (e.g., knowledge of a particular interface style for graphical user interfaces), end users with content or task knowledge, or other types of professionals. The different inspection methods have slightly different goals, but normally usability inspection is intended as a way of evaluating user interface designs to find usability problems. In usability inspection, the evaluation of the user interface is

based on the considered judgment of the inspector(s). The individual inspection methods vary as to how this judgment is derived and on what evaluative criteria inspectors are expected to base their judgments. In general, the defining characteristic of usability inspection is the reliance on judgment as a source of evaluative feedback on specific elements of a user interface. See the appendix for a short summary of the individual usability inspection methods discussed in this paper.

Usability inspection methods were first described in formal presentations in 1990 at the CHI'90 conference where papers were published on heuristic evaluation (Nielsen and Molich, 1990) and cognitive walkthroughs (Lewis et al., 1990). Now, only four to five years later, usability inspection methods have become some of the most widely used methods in the industry. As an example, in his closing plenary address at the Usability Professionals' Association's annual meeting in 1994 (UPA'94), Ken Dye, usability manager at Microsoft, listed the four major recent changes in Microsoft's approach to usability as:

- Use of heuristic evaluation
- Use of "discount" user testing with small sample sizes
- Contextual inquiry
- Use of paper mock-ups as low-fidelity prototypes

Many other companies and usability consultants are also known to have embraced heuristic evaluation and other inspection methods in recent years. Here is an example of an email message I received from one consultant in August 1994:

> "I am working [...] with an airline client. We have performed so far, 2 iterations of usability [...], the first being a heuristic evaluation. It provided us with tremendous information, and we were able to convince the client of its utility [...]. We saved them a lot of money, and are now ready to do a full lab usability test in 2 weeks. Once we're through that, we may still do more heuristic evaluation for some of the finer points."

Work on the various usability inspection methods obviously started several years before the first formal conference presentations. Even so, current use of heuristic evaluation and other usability inspection methods is still a remarkable example of rapid technology transfer from research to practice over a period of very few years.

Technology Transfer

There are many characteristics of usability inspection methods that would seem to help them achieve rapid penetration in the "marketplace of ideas" in software development organizations:

- Many companies have just recently realized the urgent need for increased usability activities to improve their user interfaces. Since usability inspection methods are cheap to use and do not require special equipment or lab facilities, they may be among the first methods tried.

- The knowledge and experience of interface designers and usability specialists need to be broadly applied; inspections represent an efficient way to do this. Thus, inspections serve a similar function to style guides by spreading the expertise and knowledge of a few to a broader audience, meaning that they are well suited for use in the many companies that have a much smaller number of usability specialists than needed to provide full service to all projects.
- Usability inspection methods present a fairly low hurdle to practitioners who want to use them. In general, it is possible to start using simple usability inspection after a few hours of training. Also, inspection methods can be used in many different stages of the system development lifecycle.
- Usability inspection can be integrated easily into many established system development practices; it is not necessary to change the fundamental way projects are planned or managed in order to derive substantial benefits from usability inspection.
- Usability inspection provides instant gratification to those who use it; lists of usability problems are available immediately after the inspection and thus provide concrete evidence of aspects of the interface that need to be improved.

To further study the uptake of new usability methods, I conducted a survey of the technology transfer of usability inspection methods.

Method

The data reported in the following was gathered by surveying the participants in a course on usability inspection taught in April 1993. A questionnaire was mailed to all 85 regular attendees in the tutorial taught by the author at the INTERCHI'93 conference in Amsterdam. Surveys were not sent to students under the assumption that they would often not be working on real projects and that they therefore could not provide representative replies to a technology transfer survey. Similarly, no questionnaires were sent to instructors from other INTERCHI'93 tutorials who were sitting in on the author's tutorial, since they were deemed to be less representative of the community at large.

Of the 85 mailed questionnaires, 4 were returned by the post office as undeliverable, meaning that 81 course attendees actually received the questionnaire. 42 completed questionnaires were received, representing a response rate of 52%.

The questionnaire was mailed in mid-November 1993 (6.5 months after the tutorial) with a reminder mailed in late December 1993 (8 months after the tutorial). 21 replies were received after the first mailing, and another 21 replies were received after the second mailing. The replies thus reflect the respondents' state approximately seven or eight months after the tutorial.

With a response rate of 49%, it is impossible to know for sure what the other half of the course participants would have replied if they had returned the questionnaire. However, data from the two response rounds allows us to speculate on possible differences based on the assumption that the non-respondents would be more like the second-round respondents than the first-round respondents. Table 1 compares these two groups on some relevant parameters. The first conclusion is that none of the differences between the

groups are statistically different, meaning that it is likely that the respondents are fairly representative of the full population. Even so, there might be a slight tendency to having the respondents were associated with larger projects than the non-respondents and that the respondents were probably more experienced with respect to usability methods than the non-respondents. Thus, the true picture with respect to the full group of tutorial participants is might reflect slightly less usage of the usability inspection methods than reported here but probably not much less.

Question	First-round Respondents	Second-round Respondents	*p*
Usability effort on project in staff-years	3.1	1.3	.2
Had used user testing before the course	89%	70%	.1
Had used heuristic evaluation after the course	65%	59%	.7
Number of different inspection methods used after course	2.2	1.8	.5

Table 1
Comparison of respondents from the first questionnaire round with the respondents from the second round. None of the differences between groups are statistically significant.

The median ratio between the usability effort of the respondents' latest project and the project's size in staff-year was 7%. Given the sample sizes, this is equivalent to the 6% of development budgets that was found to be devoted to usability in 31 projects with usability engineering efforts in a survey conducted in January 1993 (Nielsen, 1993). This result further adds to the speculation that our respondents are reasonably representative.

Questionnaire Results

Respondents were asked which of the inspection methods covered in the course they had used in the (approximately 7-8 month) period after the course. They were also asked whether they had conducted user testing after the course. The results from this question are shown in Table 2. Usage frequency in a specific period may be the best measure of the fit between the methods and project needs since it is independent of the methods' history. User testing and heuristic evaluation were clearly used much more than the other methods.

Method	Respondents Using Method After INTERCHI	Times Respondents Had Used the Method (Whether Before or After the Course)	Mean Rating of Benefits from Using Method
User testing	55%	9.3	4.8
Heuristic evaluation	50%	9.1	4.5
Feature inspection	31%	3.8	4.3

Heuristic estimation	26%	8.3	4.4
Consistency inspection	26%	7.0	4.2
Standards inspection	26%	6.2	3.9
Pluralistic walkthrough	21%	3.9	4.0
Cognitive walkthrough	19%	6.1	4.1

Table 2
Proportion of the respondents who had used each of the inspection methods and user testing in the 7-8 month period after the course, the number of times respondents had used the methods, and their mean rating of the usefulness of the methods on a 1-5 scale (5 best). Methods are sorted by frequency of use after the course.

Respondents were also asked how many times they had used the methods so far, whether before or after the course. Table 2 shows the mean number of times each method had been used by those respondents who had used it at all. This result is probably a less interesting indicator of method usefulness than is the proportion of respondents who had used the methods in the fixed time interval after the course, since it depends on the time at which the method was invented: older methods have had time to be used more than newer methods.

Finally, respondents were asked to judge the benefits of the various methods for their project(s), using the following 1-5 scale:

1 = completely useless
2 = mostly useless
3 = neutral
4 = somewhat useful
5 = very useful

The results from this question are also shown in Table 2. Respondents were only rated those methods with which they had experience, so not all methods were rated by the same number of people. The immediate conclusion from this question is that all the methods were judged useful, getting ratings of at least 3.9 on a scale where 3 was neutral.

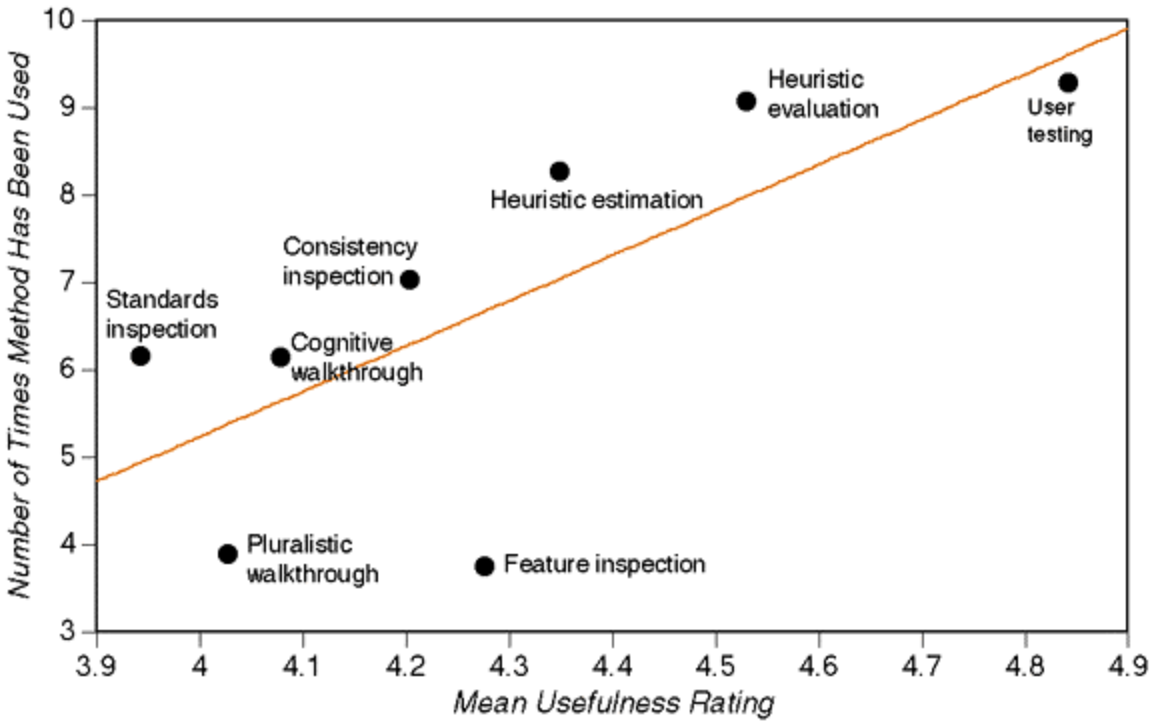

Figure 1

Regression chart showing the relation between the rated usefulness of each method and the number of times those respondents who had tried a method had used it. Data was only given by respondents who had tried a method.

The statistics for proportion of respondents having used a method, their average usefulness rating of a method, and the average number of times they had used the method were all highly correlated. This is only to be expected, as people would presumably tend to use the most useful methods the most. Figure 1 shows the relation between usefulness and times a method was used ($r = .71$, $p < .05$) and Figure 2 shows the relation between usefulness and the proportion of respondents who had tried a method whether before or after the course ($r = .85$, $p < .01$). Two outliers were identified: Feature inspection had a usefulness rating of 4.3 which on the regression line would correspond to being used 6.7 times though in fact it had only been used 3.8 times on the average by those respondents who had used it. Also, heuristic estimation had a usefulness rating which on the regression line would correspond to having been tried by 56% even though it had in fact only been used by 38%. These two outliers can be explained by the fact that these two methods are the newest and least well documented of the inspection methods covered in the course.

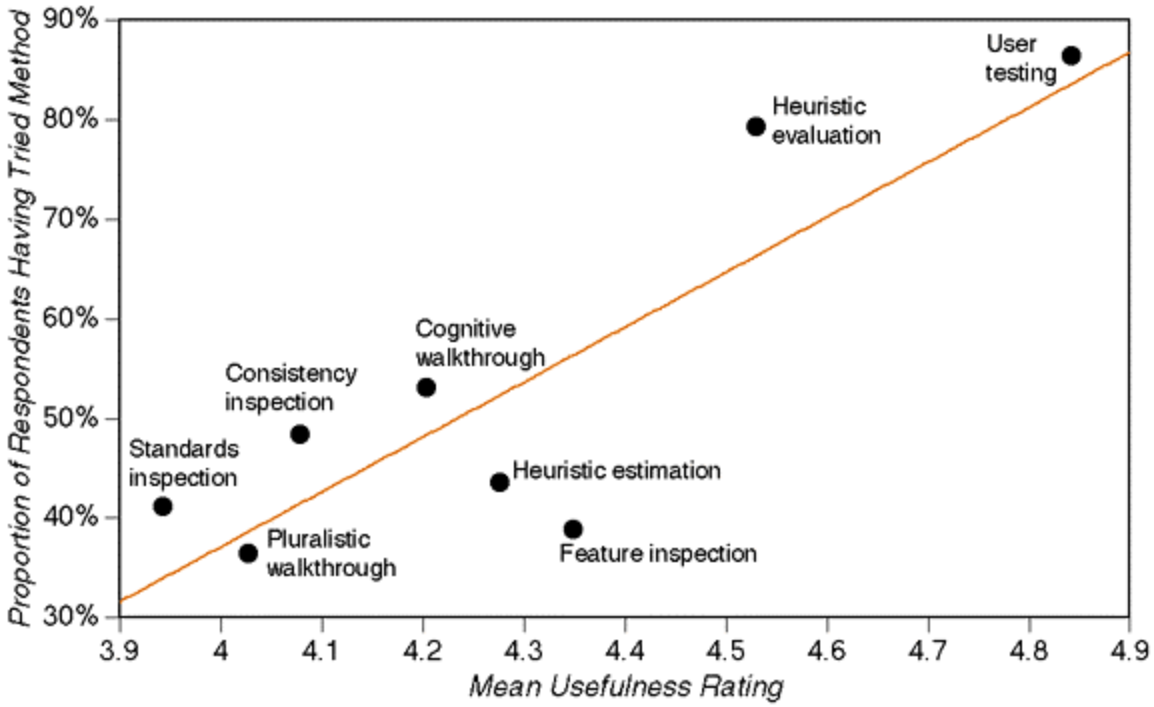

Figure 2

Regression chart showing the relation between the rated usefulness of each method and the proportion of respondents who had tried the method. Usefulness ratings were only given by those respondents who had tried a method.

The figures are drawn to suggest that usage of methods follows from their usefulness to projects. One could in fact imagine that the respondents rated those methods the highest that they had personally used the most in order to avoid cognitive dissonance, meaning that causality worked in the opposite direction as that implicitly shown in the figures. However, the correlation between the individual respondents' ratings of the usefulness of a method and the number of times they had used the method themselves is very low (r=.05), indicating that the respondents judged the usefulness of the methods independently of how much they had used them personally. There is only a high correlation in the aggregate between the mean values for each method. Thus, we conclude that the reason for this high correlation is likely to be that usability methods are used more if they are judged to be of benefit to the project. This is not a surprising conclusion but it does imply that inventors of new usability methods will need to convince usability specialists that their methods will be of benefit to concrete development projects.

Method	Respondents using the method as it was taught
Pluralistic walkthrough	27%
Heuristic estimation	25%
Heuristic evaluation	24%
Standards inspection	22%
Cognitive walkthrough	15%
Feature inspection	12%
Consistency inspection	0%

Table 3
Proportion of respondents who used the methods the way they were taught. For each method, the proportion is computed relative to those respondents who had used the method at least once.

The survey showed that only 18% of respondents used the methods the way they were taught. 68% used the methods with minor modifications, and 15% used the methods with major modifications (numbers averaged across methods). In general, as shown in Table 3, the simpler methods seemed to have the largest proportion of respondents using them as they were taught. Of course, it is perfectly acceptable for people to modify the methods according to their specific project needs and the circumstances in their organization. The high degree of method modification does raise one issue with respect to research on usability methodology, in that one cannot be sure that different projects use the "same" methods the same way, meaning that one will have to be careful when comparing reported results.

The normal recommendation for heuristic evaluation is to use 3-5 evaluators. Only 35% of the respondents who used heuristic evaluation did so, however. 38% used two evaluators and 15% only used a single evaluator. The histogram in Figure 3 shows the distribution of number of evaluators used for heuristic evaluation.

With respect to user testing, even though 35% did use 3-6 test participants (which would normally be referred to as discount usability testing), fully 50% of the respondents used 10 participants or more. Thus, "deluxe usability testing" is still being used to a great extent. The histogram in Figure 4 shows the distribution of number of test participants used for a test.

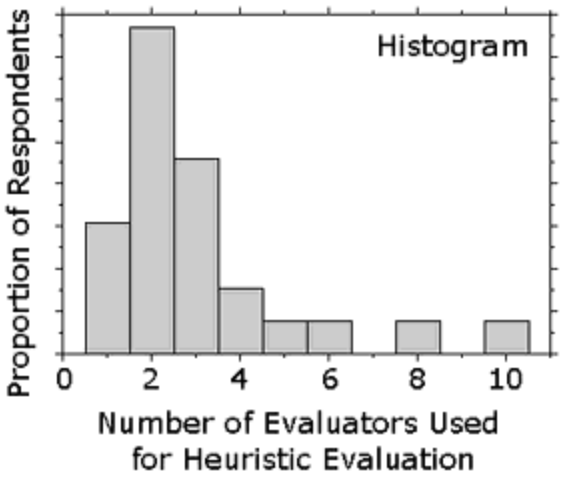

Figure 3
Histogram of the number of evaluators normally used by the respondents for heuristic evaluations.

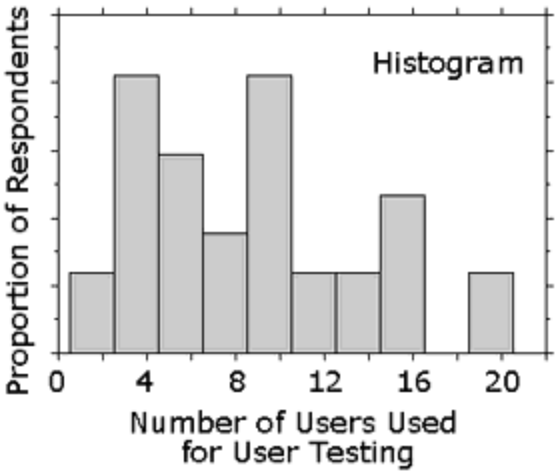

Figure 4
Histogram of the number of test users normally used by the respondents for user testing.

As one might have expected, the participants' motivation for taking the course had major impact on the degree to which they actually used the inspection methods taught in the course. People who expected to need the methods for their current project indeed did use the methods more than people who expected to need them for their next project, who again used more methods than people who did not anticipate any immediate need for the methods. Table 4 shows the number of different inspection methods used in the (7-8 month) period after the course for participants with different motivation. The table also shows the number of inspection methods planned for use during the next six months. Here, the participants with pure academic or intellectual interests have the most ambitions plans, but we still see that people who had the most immediate needs when they originally took the course plan to use more methods than people who had less immediate needs.

Motivation for taking the course	Proportion of the respondents	Number of different inspection methods used since the course	Number of different inspection methods planned for use during the next six months
Specific need to know for current project	31%	3.0	2.2
Expect to need to know for next project	21%	1.4	1.7
Expect the topic to be important in future, but don't anticipate any immediate need	14%	1.2	1.3
Pure academic or intellectual interest	12%	2.0	3.4

Table 4
Relation between the main reason people took the course and the number of different methods they have used.

In addition to the reasons listed in Table 4, 22% of the respondents indicated other reasons for taking the course. 5% of the respondents wanted to see how the instructor presented the materials in order to get material for use in their own classes and 5% wanted to validate their own experience with usability inspection and/or were developing new inspection methods. The remaining 12% of the respondents were distributed over a variety of other reasons for taking the course, each of which was only given by a single respondent.

Free-Form Comments

At the end of the questionnaire, respondents were asked to state their reasons for using or not using the various methods. A total of 186 comments were collected, comprising 119 reasons why methods were used and 67 reasons why methods were not used.

	Cognitive walkthrough	Consistency inspection	Feature inspection	Heuristic evaluation	Heuristic estimation	Pluralistic walkthrough	Standards inspection	User testing	F c
Method generates good/bad information	9 / 1	5 / 0	5 / 0	3 / 1	4 / 2	5 / 0	6 / 0	20 / 0	
Resource and/or time requirements	1 / 3	1 / 3	4 / 1	8 / 1	1 / 2	0 / 11	1 / 0	0 / 2	
Expertise and/or skills required	1 / 8	1 / 3	0 / 4	5 / 1	0 / 3		1 / 4		
Specific characteristics of individual project	2 / 0	2 / 4	1 / 2		2 / 1		0 / 6	1 / 0	
Communication, team-building, propaganda		2 / 0	1 / 0		3 / 0	5 / 0		4 / 0	
Method mandated by management		1 / 0	1 / 0	1 / 0	1 / 0		1 / 0	2 / 0	
Interaction between multiple methods				3 / 0	1 / 0	1 / 0	0 / 1		
Other reasons	0 / 2			2 / 0					
Proportion of comments that were positive	48%	55%	63%	88%	60%	50%	45%	93%	

Table 5

Classification of the 186 free-form comments made by respondents when asked to explain why they used (or did not method. In each cell, the first number indicates reasons given for using a method and the second number (after the indicates reasons given for *not* using a method (empty cells indicate that nobody made a comment about a method category)

Table 5 summarizes the free-form comments according to the following categories:

- Method generates good/bad information: reasons referring to the extent to which the results of using a method are generally useful.
- Resource and/or time requirements: reasons related to the expense and time needed to use a method.
- Expertise and/or skills required: reasons based on how easy or difficult it is to use a method. Mostly, positive comments praise methods for being easy and approachable and negative comments criticize methods for being too difficult to learn. One exception was a comment that listed it as a reason to use heuristic evaluation that it allowed usability specialists to apply their expertise.
- Specific characteristics of individual project: reasons referring to why individual circumstances made a method attractive or problematic for a specific project. For example, one comment mentioned that there was no need for consistency inspection in a project because it was the first GUI in the company and thus did not have to be consistent with anything.
- Communication, team-building, propaganda: reasons referring to the ways in which use of a method helps evangelize usability, generate buy-in, or simply placate various interest groups.
- Method mandated by management: reasons mentioning that something was done because it was a requirement in that organization.
- Interaction between multiple methods: reasons referring to the way the specific method interacts with or supplements other usability methods.

It can be seen from Table 5 that the most important attribute of a usability method is the quality of the data it generates and that user testing is seen as superior in that respect. In other words, for a new usability method to be successful, it should first of all be able to generate useful information.

The two following criteria in the table are both related to the ease of using the methods: resources and time as well as expertise and skill needed. The respondents view heuristic evaluation as superior in this regard and express reservations with respect to cognitive walkthroughs and pluralistic walkthroughs. Remember that the survey respondents came from projects that had already decided to use usability engineering and that had invested in sending staff to an international conference. The situation in many other organizations is likely to make the cost and expertise issues even more important elsewhere.

Conclusions

In planning for technology transfer of new usability methods, we have seen that the first requirement is to make sure that the method provides information that is useful in making user interfaces better. Equally important, however, is to make the method cheap and fast to use and to make it easy to learn. Actually, method proponents should make sure to cultivate the impression that their method is easy to learn since decisions as to

what methods to use are frequently made based on the method's reputation, and not by assessing actual experience from pilot usage. It is likely that cognitive walkthrough suffers from an image problem due to the early, complicated, version of the method (Lewis et al., 1990), even though recent work has made it easier to use (Wharton et al., 1994). The need for methods to be cheap is likely to be even stronger in the average development projects than in those represented in this survey, given that they were found to have above-average usability budgets.

Furthermore, methods should be flexible and able to adapt to changing circumstances and the specific needs of individual projects. The free-form comments analyzed in Table 5 show project needs as accounting for 11% of the reasons listed for use or non-use of a method, but a stronger indication of the need for adaptability is the statistic that only 18% of respondents used the methods the way they were taught, whereas 68% required minor modifications and 15% required major modifications.

A good example of flexibility is the way heuristic evaluation can be used with varying numbers of evaluators. The way the method is usually taught (Nielsen, 1994a) requires the use of 3-5 evaluators who should preferably be usability specialists. Yet, as shown in Figure 3, many projects were able to use heuristic evaluation with a smaller number of evaluators. Of course, the results will not be quite as good, but the method exhibits "graceful degradation" in the sense that small deviations from the recommended practice only results in slightly reduced benefits.

The survey very clearly showed that the way to get people to use usability methods is to get to them at the time when they have specific needs for the methods on their current project (Table 4). This finding again makes it easier to transfer methods that have wide applicability across a variety of stages of the usability lifecycle. Heuristic evaluation is a good example of such a method since it can be applied to early paper mock-ups or written specifications as well as later prototypes, ready-to-ship software, and even the clean-up of legacy mainframe screens that need to be used for a few more years without available funding for major redesign.

A final issue in technology transfer is the need for aggressive advocacy. Figure 1 shows that heuristic evaluation is used somewhat more than its rated utility would justify and that feature inspection is used much less that it should be. The most likely reason for this difference is that heuristic evaluation has been the topic of many talks, panels, seminars, books, and even satellite TV shows (Shneiderman, 1993) over the last few years, whereas feature inspection has had no vocal champions in the user interface community.

Acknowledgments

I thank Michael Muller for help in developing the survey and the many anonymous respondents for taking the time to reply. I thank Robin Jeffries and Michael Muller for helpful comments on an earlier version of this manuscript.

References

- Bell, B. (1992). Using programming walkthroughs to design a visual language. Technical Report CU-CS-581-92 (Ph.D. Thesis), University of Colorado, Boulder, CO.
- Bias, R. G. (1994). The pluralistic usability walkthrough: Coordinated empathies. In Nielsen, J., and Mack, R. L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, 65-78.
- Kahn, M. J., and Prail, A. (1994). Formal usability inspections. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, 141-172.
- Lewis, C., Polson, P., Wharton, C., and Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. Proceedings ACM CHI'90 Conference (Seattle, WA, April 1-5), 235-242.
- Nielsen, J. (1993). Usability Engineering (revised paperback edition 1994). Academic Press, Boston.
- Nielsen, J. (1994a). Heuristic evaluation. In Nielsen, J., and Mack, R. L. (Eds.), Usability Inspection Methods. John Wiley & Sons, New York. 25-62.
- Nielsen, J. (1994b). Enhancing the explanatory power of usability heuristics. Proceedings ACM CHI'94 Conference (Boston, MA, April 24-28), 152-158.
- Nielsen, J., and Mack, R. L. (Eds.) (1994). Usability Inspection Methods. John Wiley & Sons, New York.
- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces. Proc. ACM CHI'90 (Seattle, WA, April 1-5), 249-256.
- Nielsen, J., and Phillips, V. L. (1993). Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared. Proceedings ACM/IFIP INTERCHI'93 Conference (Amsterdam, The Netherlands, April 24-29), 214-221.
- Shneiderman, B. (Host) (1993). User Interface Strategies '94. Satellite TV show and subsequent videotapes produced by the University of Maryland's Instructional Television System, College Park, MD.
- Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In Nielsen, J., and Mack, R. L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, 105-140.
- Wixon, D., Jones, S., Tse, L., and Casaday, G. (1994). Inspections and design reviews: Framework, history, and reflection. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, 79-104.