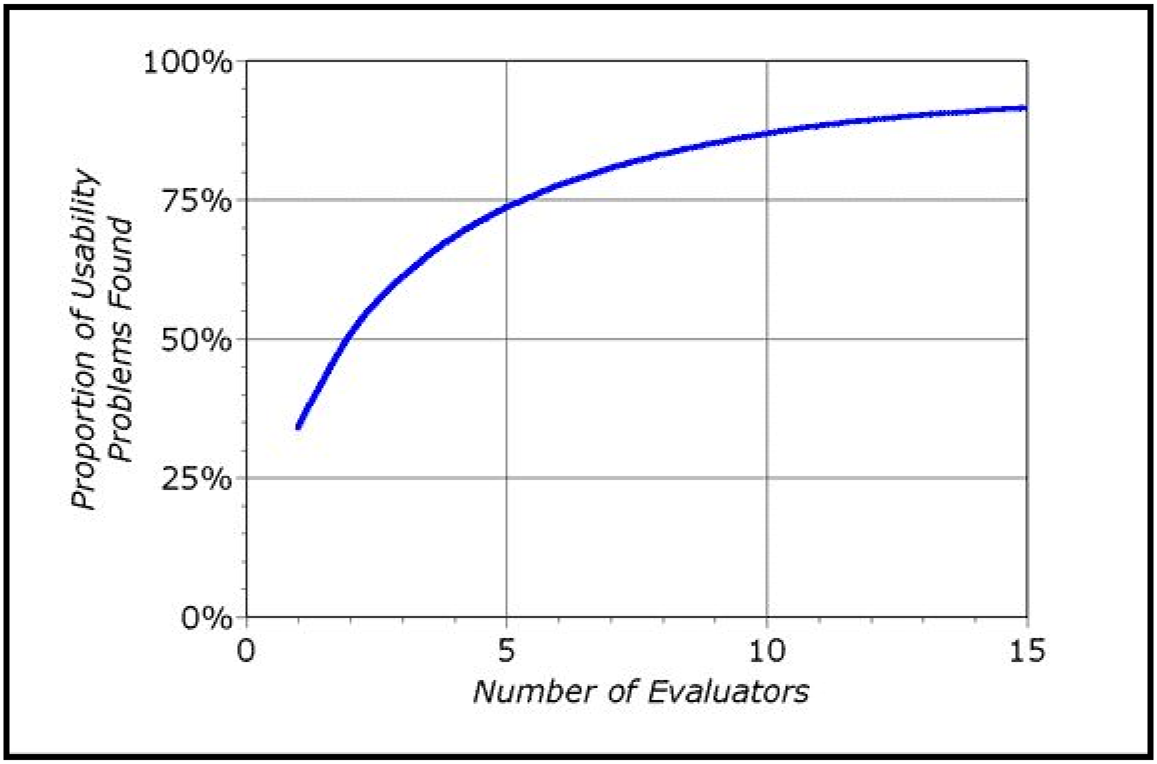
Lecture Notes:

- **Cognitive Walkthrough:**
- **Cognitive walkthroughs** involve simulating a user's problem-solving process at each step checking to see if the user's goals and memory for action can be used to lead to the next correct action.
- **Heuristic Evaluation:**
- Nielsen's Heuristics:
    1. Visibility of system status
    2. Match between system & real world
    3. User control & freedom
    4. Consistency & standards
    5. Error prevention
    6. Recognition rather than recall
    7. Flexibility and efficiency of use
    8. Aesthetic and minimalist design
    9. Help users recognize, diagnose, & recover from errors
    10. Help and documentation
- More evaluators find more problems. Incremental benefits of added evaluators decrease.



- **Severity Ratings:**
- Severity is a combination of:
    - **Frequency:** Common or rare problem?
    - **Impact:** Easy or difficult to overcome?
    - **Persistence:** One-time problem that can be overcome, or will it repeatedly bother users?
    - **Market impact**
- 0 = Not a usability problem

- 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

Textbook Notes:

- **How to Conduct a Heuristic Evaluation:**
- Summary: **Heuristic evaluation** involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles, the **heuristics**.
- **Heuristic evaluation** 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.
- A **heuristic evaluation** is a usability inspection method for computer software that helps to identify usability problems in the user interface (UI) design. It specifically involves evaluators examining the interface and judging 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.
- 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 non-domain 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.
- These are the heuristics:
    1. **Visibility of system status**:
       The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
    2. **Match between system and the real world**:
       The system should speak the user's 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.
    3. **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.
    4. **Consistency and standards**:
       Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
    5. **Error prevention**:
       Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions

   or check for them and present users with a confirmation option before they commit to the action.

6. **Recognition rather than recall**:
   Minimize the user's memory load by making 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.
7. **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.
8. **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.
9. **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.
10. **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.

- 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 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. 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. 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.
- 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. 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.
- **Severity Ratings for Usability Problems:**
- Summary: **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 eorts. If the severity ratings indicate that several disastrous usability problems remain in an interface, it will probably be inadvisable 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 four factors:
    1. The **frequency** with which the problem occurs: Is it common or rare?
    2. The **impact** of the problem if it occurs: Will it be easy or difficult for the users to overcome?

3. 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?
4. 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
- 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 have found each problem. 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.
- **The 4 questions to ask in a cognitive walkthrough:**
- The **cognitive walkthrough** is a formalised way of imagining people's thoughts and actions when they use an interface for the first time. Walkthroughs identify problems that new users will have when they first use an interface. You select one of the tasks that the

design is intended to support and then you step through the task, action by action, seeing if you can identify any problems with the interface.

- Before you can start a cognitive walkthrough, you need a complete, written list of the actions needed to complete the task with the interface, the **happy path**. For example, here's the happy path for creating a customised voicemail message on an iPhone:
    1. Tap Voicemail.
    2. Tap Greeting.
    3. Tap Custom.
    4. Tap Record and speak your greeting.
    5. When you finish, tap Stop.
    6. To listen to your greeting, tap Play.
    7. To re-record, repeat steps 4 and 5.
    8. Tap Save.
- Once you have the happy path, you're ready to start the walkthrough.
- The cognitive walkthrough is structured around 4 questions that you ask of every step in the task. You ask these questions before, during and after each step in the happy path. If you find a problem, you make a note and then move on to the next step of the task.
- Q1: Will the customer realistically be trying to do this action?
    - This question finds problems with interfaces that make unrealistic assumptions about the level of knowledge or experience that users have. It also finds problems with systems where users expect to do a different action because of their experience with other interfaces or with life generally.
- Q2: Is the control for the action visible?
    - This question identifies problems with hidden controls, like the gestural user interfaces required by an iPad where it's not always obvious what you can do. It also highlights issues with context-sensitive menus or controls buried too deep within a navigation system. If the control for the action is non-standard or unintuitive then it will identify those as well.
    - The world of TV remote controls provides a familiar example. Remote controls often contain a flap to hide features that are rarely used. The problem occurs when you need access to those functions but, because you rarely use them, you don't realise you need to lift a flap to reveal them.
- Q3: Is there a strong link between the control and the action?
    - This question highlights problems with ambiguous or jargon terms, or with other controls that look like a better choice. It also finds problems with actions that are physically difficult to execute.
- Q4: Is feedback appropriate?
    - This question helps you find problems when feedback is missing, or easy to miss, or too brief, poorly worded, inappropriate or ambiguous. For example, does the system prompt users to take the next step in the task?
- **Indirect observation: tracking users' activities:**
- **Indirect observation: tracking users' activities:**
- Sometimes direct observation is not possible because it is obtrusive or evaluators cannot be present over the duration of the study, and so users' activities are tracked indirectly. Diaries and interaction logs are two techniques for doing this. From the records collected evaluators reconstruct what happened and look for usability and user experience problems.

- **Diaries:** Diaries provide a record of what users did, when they did it, and what they thought about their interactions with the technology. They are useful when users are scattered and unreachable in person, as in many Internet and web evaluations. Diaries are inexpensive, require no special equipment or expertise, and are suitable. for long-term studies. Templates can also be created online to standardize entry format and enable the data to go straight into a database for analysis. These templates are like those used in open-ended online questionnaires. However, diary studies rely on participants being reliable and remembering to complete them, so incentives are needed and the process has to be straightforward and quick. Another problem is that participants often remember events as being better or worse than they really were, or taking more or less time than they actually did.
- **Interaction logging:** Interaction logging in which key presses, mouse or other device movements are recorded has been used in usability testing for many years. Collecting this data is usually synchronized with video and audio logs to help evaluators analyze users' behavior and understand how users worked on the tasks they set. Specialist software tools are used to collect and analyze the data. The log is also time-stamped so it can be used to calculate how long a user spends on a particular task or lingered in a certain part of a website or software application. An advantage of logging user activity is that it is unobtrusive. Another advantage is that large volumes of data can be logged automatically. However, powerful tools are needed to explore and analyze this data quantitatively and qualitatively. An increasing number of visualization tools are being developed for this purpose.
- **Analyzing, interpreting, and presenting the data:**
- Most observational evaluations generate a lot of data in the form of notes, sketches, photographs, audio and video records of interviews and events, various artifacts, diaries, and logs. Most observational data is qualitative and analysis often involves interpreting what users were doing or saying by looking for patterns in the data. Sometimes qualitative data is categorized so that it can be quantified and in some studies events are counted.
- Dealing with large volumes of data, such as several hours of video, is daunting, which is why it is particularly important to plan observation studies very carefully before starting them. The DECIDE framework suggests identifying goals and questions first before selecting techniques for the study, because the goals and questions help determine which data is collected and how it will be analyzed.
- When analyzing any kind of data, the first thing to do is to "eyeball" the data to see what stands out. Are there patterns or significant events? Is there obvious evidence that appears to answer a question or support a theory? Then proceed to analyze it according to the goals and questions. The discussion that follows focuses on three types of data:
    1. **Qualitative data that is interpreted and used to tell a story about what was observed.** Much of the power of analyzing descriptive data lies in being able to tell a convincing story, illustrated with powerful examples that help to confirm the main points and will be credible to the development team. It is hard to argue with well-chosen video excerpts of users interacting with technology or anecdotes from transcripts. To summarize, the main activities involved in working with qualitative data to I tell a story are:
        - Review the data after each observation session to synthesize and identify key themes and make collections.

- Record the themes in a coherent yet flexible form, with examples. While post-its enable you to move ideas around and group similar ones, they can fall off and get lost and are not easily transported, so capture the main points in another form, either on paper or on a laptop, or make an audio recording.
- Record the date and time of each data analysis session. (The raw data should already be systematically logged with dates.)
- As themes emerge, you may want to check your understanding with the people you observe or your informants.
- Iterate this process until you are sure that your story faithfully represents what you observed and that you have illustrated it with appropriate examples from the data.
- Report your findings to the development team, preferably in an oral presentation as well as in a written report. Reports vary in form, but it is always helpful to have a clear, concise overview of the main findings presented at the beginning.

2. **Qualitative data that is categorized using techniques such as content analysis.** Data from think-aloud protocols, video, or audio transcripts can be analyzed in different ways. These can be coarse-grained or detailed analyses of excerpts from a protocol in which each word, phrase, utterance, or gesture is analyzed. Sometimes examining the comment or action in the context of other behavior is sufficient. Some are used more often in research while others are used more for product development. Analyzing even a short half-hour videotape would be very time-consuming if evaluators studied every comment or action in detail. Furthermore, such fine-grained analyses are often not necessary. A common strategy is to look for critical incidents, such as times when users were obviously stuck. Such incidents are usually marked by a comment, silence, looks of puzzlement, etc. Evaluators focus on these incidents and review them in detail, using the rest of the video as context to inform their analysis.
3. **Quantitative data that is collected from interaction and video logs and presented as values, tables, charts and graphs and is treated statistically.** Content analysis provides another fine grain way of analyzing video data. It is a systematic, reliable way of coding content into a meaningful set of mutually exclusive categories. The content categories are determined by the evaluation questions and one of its most challenging aspects is determining meaningful categories that are orthogonal. I.e. They do not overlap each other in any way. Deciding on the appropriate granularity is another issue to be addressed. The content categories must also be reliable so that the analysis can be replicated. This can be demonstrated by training a second person to use the categories. When training is complete, both researchers analyze the same data sample. If there is a large discrepancy between the two analyses, either training was inadequate or the categorization is not working and needs to be refined. By talking to the researchers you can determine the source of the problem, which is usually with the categorization. If so, then a better categorization scheme needs to be devised and re-tested by doing more inter-researcher reliability tests. However, if the researchers do not seem to know how to carry out the process then they probably need more training. When a high level of reliability is reached, it can be quantified by calculating an inter-research reliability rating. This is the

percentage of agreement between the two researchers, defined as the number of items that both categorized in the same way expressed as a percentage of the total number of items examined. It provides a measure of the efficacy of the technique and the categories. Another approach to video, and audio analysis is to focus on the dialog, i.e., the meaning of what is said, rather than the content. Discourse analysis is strongly interpretive, pays great attention to context, and views language not only as reflecting psychological and social aspects but also as constructing it. An underlying assumption of discourse analysis is that there is no objective scientific truth. Language is a form of social reality that is open to interpretation from different perspectives. Video data collected in usability laboratories is usually annotated as it is observed. Small teams of evaluators watch monitors showing what is being recorded in a control room out of the users' sight. As they see errors or unusual behavior, one of the evaluators marks the video and records a brief remark. When the test is finished evaluators can use the annotated recording to calculate performance times so they can compare users' performance on different prototypes. The data stream from the interaction log is used in a similar way to calculate performance times. Typically this data is further analyzed using simple statistics such as means, standard deviations, T-tests, etc. Categorized data may also be quantified and analyzed statistically, as we have said.

- The results from an evaluation can be reported to the design team in several ways, as we have indicated. Clearly written reports with an overview at the beginning and detailed content list make for easy reading and a good reference document. Including anecdotes, quotations, pictures, and video clips helps to bring the study to life, stimulate interest, and make the written description more meaningful. Some teams like quantitative data, but its value depends on the type of study and its goals. Verbal presentations that include video clips can also be very powerful. Often both qualitative and quantitative data analysis are useful because they provide alternative perspectives.