7.HCI Evaluation 人机交互评估

Key point:

Think Aloud technique and Heuristic Evaluation

HCI (Human-Computer Interaction) evaluation is the process of assessing the usability, efficiency, and effectiveness of a computer system or software application. It involves gathering data on how users interact with a system or application and analyzing this data to identify any issues or areas for improvement.HCI（人机交互）评估是评估计算机系统或软件应用程序的可用性、效率和有效性的过程。它涉及收集有关用户如何与系统或应用程序交互的数据，并分析这些数据以确定任何问题或需要改进的地方。

There are several methods for conducting HCI evaluation, including:

1. Usability testing可用性测试: This involves observing users as they complete specific tasks with the system or application, and collecting data on their performance and user experience.这涉及观察用户使用系统或应用程序完成特定任务，并收集有关其性能和用户体验的数据。
2. Heuristic evaluation启发式评估: This involves having experts evaluate the system or application based on a set of established usability principles, or heuristics.这涉及让专家根据一组已建立的可用性原则或启发式评估系统或应用程序
3. Cognitive walkthrough 认知演练: This involves walking through the system or application from the perspective of a user, and identifying any potential issues with the interface or user flow.这涉及从用户的角度浏览系统或应用程序，并识别界面或用户流程的任何潜在问题
4. User surveys 用户调查: This involves gathering feedback from users through questionnaires or surveys to identify their opinions and preferences regarding the system or application.这涉及通过问卷或调查收集用户的反馈，以确定他们对系统或应用程序的意见和偏好。

The goal of HCI evaluation is to identify any usability issues with the system or application and make improvements to enhance the user experience. This can lead to increased user satisfaction, improved efficiency, and higher productivity.HCI 评估的目标是识别系统或应用程序的任何可用性问题，并进行改进以增强用户体验。这可以提高用户满意度、提高效率和生产力。

Think Aloud technique and Heuristic Evaluation are both important methods in evaluating user interfaces and ensuring that software meets user requirements. 有声思考技术和启发式评估都是评估用户界面和确保软件满足用户需求的重要方法。

The Think Aloud technique involves asking users to verbalize their thoughts and actions as they interact with a system, providing valuable insight into their thought processes and revealing any usability issues.Think Aloud 技术涉及要求用户在与系统交互时用语言表达他们的想法和行动，提供对他们思维过程的有价值的洞察并揭示任何可用性问题。

Heuristic Evaluation, on the other hand, involves expert evaluators using a set of heuristics**/ hjuˈrɪstɪks /** 启发式 or guidelines to identify potential usability issues and areas for improvement in a system. Both methods can be used in conjunction with user testing to provide a comprehensive evaluation of a system's usability and user experience. It is recommended that these methods are carried out as part of the user-centred design process in the development of software, including game development. The results of these evaluations can then be used to inform improvements and changes to the user interface and overall design of the system. 另一方面，启发式评估涉及专家评估人员使用一组启发式或指南来识别系统中潜在的可用性问题和需要改进的领域。这两种方法都可以与用户测试结合使用，以提供对系统可用性和用户体验的全面评估。建议将这些方法作为软件开发（包括游戏开发）中以用户为中心的设计过程的一部分来执行。

Why is evaluation important?

Iterative design, with its repeating cycle of design and testing, is the only validated methodology in existence that will consistently produce successful results. If you don’t have user-testing as an integral part of your design processyou are going to throw buckets of money down the drain.”

迭代设计及其设计和测试的重复循环，是现存唯一经过验证的方法，能够始终如一地产生成功的结果。如果你没有把用户测试作为你设计过程中不可或缺的一部分，你就会把一大笔钱付诸东流。

——

Bruce Tognazzini

Evaluation is important in HCI because it allows designers and developers to test and refine their designs to ensure they meet the needs and preferences of users. By testing designs with real users, we can identify usability issues and areas for improvement before a product is released. This helps to minimize the risk of user dissatisfaction and increase the chances of success in the market. Additionally, evaluation can help to ensure that the product is accessible to users with different abilities and preferences, and can identify potential ethical issues with the design. Ultimately, evaluation is important because it helps to ensure that the final product meets the needs of its users and is successful in the marketplace.

评估在 HCI 中很重要，因为它允许设计人员和开发人员测试和改进他们的设计，以确保他们满足用户的需求和偏好。通过与真实用户一起测试设计，我们可以在产品发布之前确定可用性问题和需要改进的地方。这有助于将用户不满意的风险降至最低，并增加在市场上取得成功的机会。此外，评估有助于确保产品可供具有不同能力和偏好的用户使用，并且可以识别设计中潜在的道德问题。最后，评估很重要，因为它有助于确保最终产品满足用户的需求并在市场上取得成功。

The Think Aloud evaluation technique

From slides:

●Users are asked to verbalise what they are thinking and doing as they perform a task using your software

●The Think Aloud technique provides insights into the user experience of using your software

●It can identify issues with the software e.g.navigation problems or content that can be improved

●It can be used as part of the software development process to iteratively improve software or used with a finished product

Introduction

The Think Aloud evaluation technique is a user testing method where participants are asked to verbalize their thoughts and actions as they perform a task using a piece of software. The aim of this technique is to provide insights into the user experience of using the software, to identify issues and areas for improvement, and to inform iterative design and development.Think Aloud 评估技术是一种用户测试方法，要求参与者在使用软件执行任务时用语言表达他们的想法和行动。该技术的目的是深入了解使用软件的用户体验，确定问题和需要改进的领域，并为迭代设计和开发提供信息。

During a Think Aloud test, the participant is asked to complete a specific task while continuously vocalizing their thoughts and actions. The moderator may prompt the participant with open-ended questions, but should generally avoid leading or guiding them. The session is recorded for later analysis and review.在大声思考测试中，参与者被要求完成一项特定任务，同时不断说出他们的想法和行动。主持人可以用开放式问题提示参与者，但通常应避免引导或指导他们。该会话被记录下来供以后分析和审查。

The Think Aloud technique is a useful and widely used evaluation method in industry because it is relatively quick, inexpensive, and can be used throughout the software development lifecycle. However, it is important to note that Think Aloud tests are limited in that they rely on self-reporting and may not uncover all issues with the software. Therefore, it is recommended to use multiple evaluation methods to gain a more complete understanding of the user experience.

Think Aloud 技术是一种有用且在行业中广泛使用的评估方法，因为它相对快速、便宜，并且可以在整个软件开发生命周期中使用。但是，请务必注意，Think Aloud 测试的局限性在于它们依赖于自我报告，可能无法发现软件的所有问题。因此，建议使用多种评估方法来更全面地了解用户体验。

Process:

●During a Think Aloud evaluation, the evaluator usually gives the user a specific task to perform using the software.
● The user is then asked to speak out loud, explaining their thought process and actions as they try to complete the task.
● The evaluator can observe the user's behavior and listen to their comments to gain insight into potential usability issues.
● The process is recorded (with consent) and the data collected can be analyzed to

identify patterns and common issues.
● Think Aloud evaluation is a relatively simple and inexpensive technique that can provide valuable feedback for improving the user experience of software.

Benefits of Think Aloud

- Cheap
- Relatively easy
- It provides insight into people's experiences as they interact with your product
- It can be carried out with low numbers of participants 少数人中
- Fits in with most software development processes 适合大多数软件发展

-It helps identify usability issues and areas for improvement
• It provides immediate feedback during the testing process 即时反馈
• It allows for observation of user behavior and understanding of user thought processes
• It can be used to identify differences in the user experience between different user groups
• It helps to validate design decisions and improve the overall user experience of the software.

Drawbacks of Think Aloud

- It relies on people verbalising thoughts and impressions, rather than objective measures 答案主观
- Participants may say what they believe to be the right answer rather than what they really think (social desirability). This can distort your results and conclusions 答案可信度
- The Think Aloud technique can disrupt the user's natural behavior, causing them to behave differently than they normally would.
- The results may not generalize well to the larger population as it is a small sample size.不适合人多的场景
- It may be difficult for participants to verbalize their thoughts, especially if they are not comfortable with the process or have difficulty expressing themselves. 难以用言语表达想法

- Think Aloud evaluations can take a significant amount of time to plan, conduct, and analyze.

Planning a Think Aloud evaluation

1. Identify the research questions: Determine what you want to learn from the evaluation. For example, you may want to know if users can easily complete a specific task or understand a certain feature of the software.
2. Select participants: Decide how many participants you want to recruit for the evaluation and what criteria they should meet (e.g. age, experience level). Try to recruit participants who are representative of your target user population.
3. Develop tasks: Create a list of tasks that you want the participants to perform while using your software. These tasks should be relevant to the research questions you identified in step 1.
4. Prepare materials: Prepare any necessary materials for the evaluation, such as a script to guide participants through the session or a consent form for them to sign.
5. Conduct the evaluation: During the evaluation, ask participants to think aloud while completing the tasks. Encourage them to verbalize their thoughts and feelings as they work through the software.
6. Analyze the data: Record and transcribe the data collected during the evaluation. Look for patterns and themes in the participants' feedback to identify areas where the software can be improved.
7. Report findings: Use the data to inform design decisions and report the findings to stakeholders. Consider creating a list of actionable recommendations to address any issues identified during the evaluation.

Carrying out a Think Aloud evaluation

-Have a facilitator to run the evaluation and one or two observers to take notes on what the user says. 需要有1-2个观察人员
-Explain to the participants how a think aloud works: they should tell you their thoughts, reactions and emotions as they occur while they are performing the task. 需要告知测试者评估如何进行
-Explain that there is no right answer and it's fine to be critical. 告知测试者没有正确答案，可以按照自己的感受回答

-Ask the participants to complete the tasks you have planned. This should be **uninterrupted** as far as possible, although the facilitator will probably need to give some prompts.尽量不要中断他们的想法
-If the user goes silent then prompt them to verbalise their thoughts by saying “what are you thinking” 给他们提示

1. Start by introducing the evaluator and explaining the purpose of the evaluation. Make the user feel comfortable and encourage them to speak their thoughts aloud.
2. Give the user a task to complete using your software. Be specific about what you want them to do and don't give them any hints or suggestions.
3. As the user performs the task, ask them to describe what they are thinking and doing. Encourage them to speak aloud and describe their thought process.
4. Make note of any issues or problems that the user encounters while performing the task. This can be anything from difficulty finding a button to confusion about the purpose of a feature.
5. Repeat the process with additional tasks and users, making note of any common issues that arise.
6. Analyze the data collected during the evaluations to identify patterns and common issues. Use this information to improve the design of your software.
7. Finally, report the results of the evaluation to your team and stakeholders, and make recommendations for improvements based on the data collected.

Analysing a Think Aloud evaluation

1. Review the recordings: Watch or listen to the recordings of each participant's Think Aloud session. Take detailed notes on their actions, comments, and feedback as they interact with the software.
2. Identify common issues: Look for patterns in the feedback and comments from the participants. Note any issues that multiple participants experienced, such as difficulty navigating a particular section of the software.
3. Categorize the issues: Group the identified issues into categories, such as navigation, content, or design. This will help you identify which areas of the software

need the most attention.
4. Prioritize the issues: Determine which issues are the most important to address based on factors such as frequency of occurrence and impact on the user experience.
5. Develop solutions: For each identified issue, brainstorm potential solutions to address the problem. Consider factors such as feasibility, impact on other areas of the software, and potential costs.
6. Implement changes: Based on the solutions developed in step 5, make the necessary changes to the software. Be sure to test the changes to ensure that they address the identified issues and do not introduce new problems.
7. Repeat the process: Conduct another Think Aloud evaluation with new participants after implementing changes to the software. This will help you determine whether the changes have been effective and identify any new issues that may have arisen.

From slides:
-Put the written notes together from both observes in to one document
-Organise the notes into meaningful categories e.g.what features helped users; what features led to problems; any additional features that users wanted.
-You can make your own meanginful categories
-Count the number of times users comment about different categories to identify the biggest issues.

–Heuristic evaluation (Jakob Nielsen)

Heuristic evaluation is a usability inspection method in which evaluators examine an interface or system to identify any problems or issues that may negatively affect the user experience. It was developed by Jakob Nielsen and Rolf Molich in the early 1990s as a quick and cost-effective way to evaluate a system's usability.

The evaluation is typically carried out by a small team of evaluators who are experts in usability and user experience design. They use a set of heuristics or guidelines, such as **Nielsen's ten usability heuristics**, to evaluate the system and identify any usability issues. The evaluation is typically done independently, with each evaluator conducting their own evaluation and then comparing their results with the other evaluators.

The results of a heuristic evaluation can be used to identify and prioritize usability issues, as well as inform design decisions and improvements. It can also be used in combination with other usability evaluation methods, such as user testing, to provide a more comprehensive evaluation of a system's usability.

What is Heuristic?

A heuristic is a problem-solving strategy or technique that helps to find a solution quickly and efficiently, often by making use of practical experience, common sense, or simple rules of thumb. In the context of user interface design, heuristic evaluation is a method of assessing the usability of a software application or website by evaluating it against a set of predefined heuristics or guidelines.

Heuristic evaluation:

-An evaluation technique conducted **without** users
-Also known as **expert** evaluationas(别名) it's sometimes carried out by external experts (sometimes by the development team) aka evaluators
-It's a type of **analytical** evaluation, that is, based on a set of principles or a model…
…rather than by observing users (which is known as **empirical** evaluation)
-It's an **inspection** method –it involves inspecting a design to find usability problems
-This involves asking whether the design complies with **usability principles**(a set of heuristics)

To add on to that, heuristic evaluation involves experts evaluating the user interface of a software product to identify usability problems. They evaluate the product against a set of heuristics or usability principles, which are generally accepted guidelines for good design. The evaluators identify usability issues, rate them according to their severity, and provide recommendations for improving the interface. The method is efficient because it can be carried out quickly and provides valuable insights into usability issues that may not have been identified otherwise.

Heuristic evaluation is widely used becasue:

-It's **cheap**(only needs a small number of evaluatorsand no specialist equipment or labs)
-Relatively **easy** to carry out (can do it after a few hours of training)
-**Instant gratification 即时满足** –lists of problems are **available immediately** after the inspection
-It **fits in** with most software development processes used in industry
-It's a very **cost effective**: benefit-cost ratio of 48: cost of $10,500; expected benefits $500,000 (Nielsen 1994).

Advantages of heuristic evaluation include:

1. Cost-effective: It requires only a small number of evaluators and no special equipment or lab, making it an affordable method for many projects. 省钱
2. Immediate feedback: The evaluation can be completed quickly, and the evaluators can provide a list of identified issues immediately.即时反应
3. Helps identify potential issues: The evaluation can help identify potential issues early in the development process, allowing for quicker and more efficient design changes. 发现潜在问题
4. Flexibility: It can be used on both finished and unfinished products, as well as on a wide variety of applications. 灵活性

Disadvantages of heuristic evaluation include:

1. Limited perspective: As the evaluation is conducted by experts, it may not identify issues that are only apparent to end-users. 观点受限, 没有用户参与
2. Limited depth: Evaluators may not have access to all necessary information, such as user data or contextual factors, which could impact the evaluation.
3. Biases: Evaluators may have their own biases and heuristics, which can impact the evaluation and lead to inaccurate results.
4. Lack of validation: The evaluation method is not validated by user feedback or data, which can impact its accuracy and usefulness.

Difference between Think aloud and heuristic evaluation

1. User involvement: Think aloud involves a user performing a task while verbally expressing their thoughts and actions, while in heuristic evaluation, experts evaluate an interface without user involvement.
2. Evaluation focus: Think aloud focuses on understanding user behavior, reactions, and thought processes while using the interface, while heuristic evaluation focuses on identifying potential usability issues based on established design principles or guidelines.
3. Methodology: Think aloud is a qualitative method that provides rich insights into user behavior and reactions, while heuristic evaluation is a more quantitative method that involves applying a set of heuristics or guidelines to evaluate an interface.
4. Time and cost: Think aloud can be time-consuming and costly due to the need for user involvement and the need to recruit and compensate participants, while heuristic evaluation is typically faster and less expensive as it does not require user involvement.
5. Skill level: Think aloud can be used with both novice and expert users, while heuristic evaluation typically requires evaluators with specialized expertise in usability and design principles.

Differene between Qualitative methods and quantitative methods

Qualitative methods **定性法**:

- Qualitative methods focus on collecting non-numerical data such as words, images, and observations.
- Data collection methods include techniques such as interviews, observations, and focus groups.
- Data analysis involves identifying patterns, themes, and relationships in the data.
- The results are typically presented in a narrative form and are often subjective in nature.

Quantitative methods **计量方法;定量法**:

- Quantitative methods focus on collecting numerical data that can be measured and analyzed statistically.
- Data collection methods include surveys, experiments, and other quantitative research designs.
- Data analysis involves statistical analysis of the data to identify patterns, relationships, and significant differences.
- The results are typically presented in numerical or graphical form and are often objective in nature.

In summary, the main difference between qualitative and quantitative methods is the type of data collected and the methods used for data analysis.

Where are the users?

●Heuristic evaluation is based on HCI researchers' extensive experience of designing and evaluating interfaces
●By focusing on users, HCI researchers learned what works and what doesn't
●Their experience is distilled into **usability principles** (a set of heuristics)
●The principles represent the findings from thousands of user studies
●They have been used for over 30 years

What are Nielsen's 10 principles of heuristic evaluation?

1. Visibility of system status - The system should always keep users informed about what is going on through appropriate feedback within a reasonable amount of time. 系统状态的可见性——系统应始终在合理的时间内通过适当的反馈让用户了解正在发生的事情。
2. Match between system and the real world - The system should use language and concepts familiar to the user, rather than technical jargon and system-oriented terms.系统与现实世界的匹配——系统应该使用用户熟悉的语言和概念，而不是技术术语和面向系统的术语。
3. User control and freedom - The system should offer users a clear and easy way to undo or redo actions and exit from undesired states.用户控制和自由——系统应该为用户提供一种清晰、简单的方法来撤消或重做操作并退出不希望的状态。

4. Consistency and standards - The system should follow platform conventions and user expectations, making sure that similar actions are presented in a consistent way throughout.一致性和标准——系统应遵循平台惯例和用户期望，确保类似的操作始终以一致的方式呈现。
5. Error prevention - The system should take measures to prevent errors from occurring in the first place, such as confirmation dialogs for dangerous actions or disabling options that are not currently available.错误预防——系统应该首先采取措施防止错误发生，例如危险操作的确认对话框或禁用当前不可用的选项。
6. Recognition rather than recall - The system should reduce the user's memory load by making objects, actions, and options visible and easy to access, rather than requiring users to recall specific information.识别而不是回忆——系统应该通过使对象、动作和选项可见并易于访问来减少用户的记忆负荷，而不是要求用户回忆特定信息。
7. Flexibility and efficiency of use - The system should offer both novice and expert users efficient ways to interact with it, such as shortcuts, accelerators, and customization options.使用的灵活性和效率——系统应该为新手和专家用户提供有效的交互方式，例如快捷方式、加速器和自定义选项。
8. Aesthetic and minimalist design - The system should avoid unnecessary or irrelevant information or decoration, in order to simplify the user interface and make it more attractive. 美学和极简主义设计——系统应避免不必要或不相关的信息或装饰，以简化用户界面并使其更具吸引力。
9. Help users recognize, diagnose, and recover from errors - The system should provide clear and concise error messages that explain the problem and suggest a solution, helping users recover from errors quickly and easily. 帮助用户识别、诊断错误并从错误中恢复——系统应提供清晰简洁的错误消息来解释问题并提出解决方案，帮助用户快速轻松地从错误中恢复。
10. Help and documentation - The system should provide easy-to-find and understandable documentation and help features, allowing users to get the assistance they need when they need it. 帮助和文档——系统应提供易于查找和理解的文档和帮助功能，使用户能够在需要时获得所需的帮助。

简化版
•feedback
•metaphor
•user control and freedom
•consistency

•error prevention
•recognition not recall
•flexible use
•minimal information
•error recognition and recovery
•help

1. Visibility of system status -feedback

#1: Visibility of system status

The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.

When users know the current system status, they learn the outcome of their prior interactions and determine next steps. Predictable interactions create trust in the product as well as the brand.

7:24

Example of Usability Heuristic #1:
You Are Here *indicators on mall maps show people where they currently are, to help them understand where to go next.*

Tips

- Communicate clearly to users what the system's state is — no action with consequences to users should be taken without informing them.
- Present feedback to the user as quickly as possible (ideally, immediately).
- Build trust through open and continuous communication.

Learn more

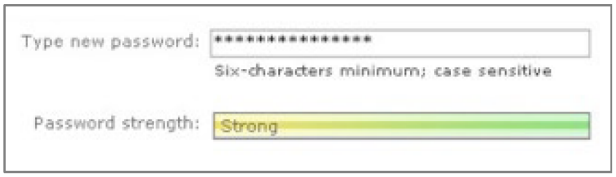
- Full article: *Visibility of System Status*
- 3-minute video about the *Visibility Heuristic*

1. Inform the user about what's going on: show appropriate feedback and progress
2. do not show blank screens
3. do not show static "load" or progress messages

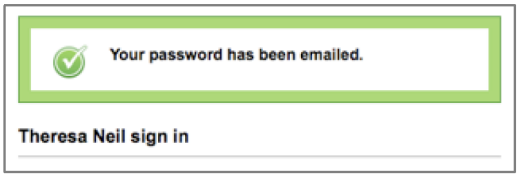
This means that the system should provide users with information about what is happening, such as whether a process is complete or still in progress, or whether an action was successful or not. It also means that the system should provide timely and relevant feedback to user actions, such as confirming that an input was received or alerting the user to errors or issues.

By providing clear and constant feedback, the user can have a better understanding of the system's behavior and can adjust their actions accordingly. This heuristic is especially important for systems with long-running processes, where the user may need to wait for some time before receiving feedback.

Visibility of system status: examples



Microsoft Live
Password strength is shown as the password is entered. Colors are used to augment the message.



Tick
A feedback message is displayed when an action is performed

Examples of Visibility of system status -feedback can include:

1. Progress bars or loading animations to indicate that the system is working on a task.
2. Success or error messages to indicate whether a task was completed or not, and if not, what went wrong.
3. Real-time updates to show changes in data or information as they occur.
4. Confirmation messages to show that a user's action has been successfully completed.

5. Navigation cues such as breadcrumb trails or menu highlighting to show where a user is within the system.

2. Match between system and real world -metaphor

#2: Match between system and the real world

The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.

7:24

The way you should design depends very much on your specific users. Terms, concepts, icons, and images that seem perfectly clear to you and your colleagues may be unfamiliar or confusing to your users.

When a design's controls follow real-world conventions and correspond to desired outcomes (called natural mapping), it's easier for users to learn and remember how the interface works. This helps to build an experience that feels intuitive.

Example of Usability Heuristic #2:
When stovetop controls match the layout of heating elements, users can quickly understand which control maps to which heating element.

Tips

- Ensure that users can understand meaning without having to go look up a word's definition.
- Never assume your understanding of words or concepts will match that of your users.
- User research will uncover your users' familiar terminology, as well as their mental models around important concepts.

Learn more

- Full article: *Match Between the System and the Real World*
- 2-minute video: *Match Between the System and the Real World*

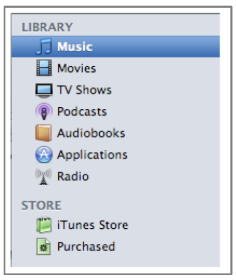
-There must be a match between the system's interface controls and the real world
-The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms
-Follow real-world conventions, making information appear in a natural and logical order

Example of Matching between system and real world -metaphor

Some examples of Match between system and real world -metaphor are:

- Using a trash can icon to represent deleting a file or item, as it is a familiar real-world object representing disposal.
- Representing a file folder with a tab and label, similar to physical file folders in the real world.
- Using a desktop metaphor to represent a computer's user interface, with a desktop, icons, folders, and trash can.
- Representing a shopping cart icon to represent a virtual shopping cart, as it is a familiar real-world object representing carrying items while shopping.

Match between system and real world - examples



iTunes
Organized as a library that contains your media library: music, movies, TV shows, audiobooks. Beneath the Library is the Store where you can buy more media to put in your Library.

3. User control and freedom - navigation

#3: User control and freedom

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.

When it's easy for people to back out of a process or undo an action, it fosters a sense of freedom and confidence. Exits allow users to remain in control of the system and avoid getting stuck and feeling frustrated.

Example of Usability Heuristic #3:
Digital spaces need quick emergency exits, just like physical spaces do.

Tips

- Support *Undo* and *Redo*.
- Show a clear way to exit the current interaction, like a *Cancel* button.
- Make sure the exit is clearly labeled and discoverable.

Learn more

- Full article: *User Control and Freedom*
- 2-minute video: *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 dialog.

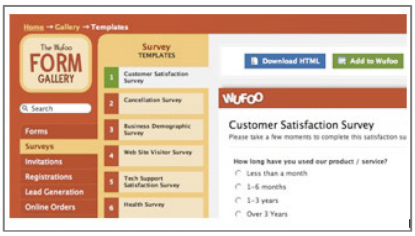
●Support undo and redo and a clear way to navigate.
●Provide bread crumbs to clearly show where the user is.

Examples of User control and freedom in navigation include:

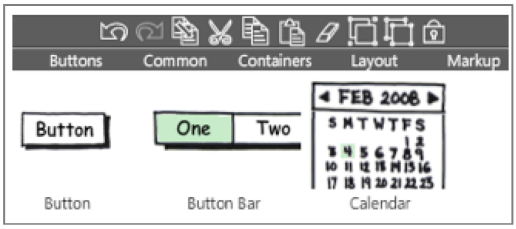
- Providing clearly marked "cancel" or "back" buttons to allow users to exit or undo an action
- Allowing users to undo and redo their actions
- Providing a clear and consistent navigation structure, such as a menu or breadcrumbs(navigation bar), to allow users to easily find their way around the system

- Providing keyboard shortcuts for frequently used functions to allow users to navigate more quickly
- Providing a search function to allow users to quickly find what they are looking for
- Allowing users to customize their navigation preferences, such as changing the order of menu items or hiding unused features

User control and freedom - examples



Wufoo
Clearly marks where the person is and where they can go by showing the selection in each menu



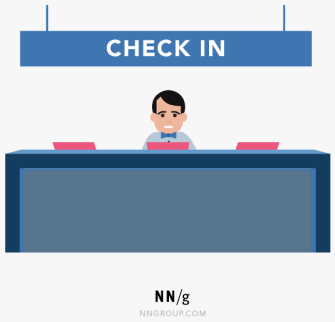
Balsamiq
Undo and Redo buttons are available in the toolbar, and can also be accessed with the standard keyboard shortcuts

4. Consistency and standards

#4: Consistency and standards

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

Jakob's Law states that people spend most of their time using digital products *other than yours*. Users' experiences with those other products set their expectations. Failing to maintain consistency may increase the users' cognitive load by forcing them to learn something new.



Example of Usability Heuristic #4:
Checkin counters are usually located at the front of hotels. This consistency meets customers' expectations.

Tips

- Improve learnability by maintaining both types of consistency: internal and external.
- Maintain consistency within a single product or a family of products (internal consistency).
- Follow established industry conventions (external consistency).

Learn more

- Full article: *Consistency and Standards*
- 3-minute video: *Consistency and Standards*

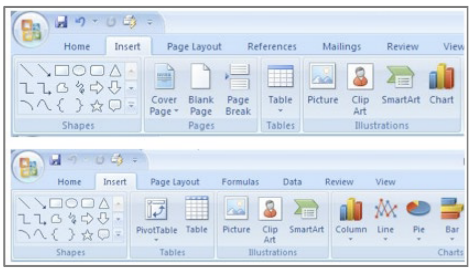
●Users should not have to wonder whether different words, situations, or actions mean the same thing.
●Follow platform conventions.

Some examples of consistency and standards include:

- Consistent placement of buttons and controls throughout the interface
- Consistent use of terminology and language across the interface
- Consistent use of visual design elements such as colors, fonts, and icons
- Consistent behavior of interface elements in response to user actions
- Consistent use of error messages and warnings.

Consistency: examples

Gmail
When Gmail was designed, they based the organizational folders on the same ones used in other client email applications: Inbox, Drafts, Sent Mail.



Microsoft Office
Word, Excel, and PowerPoint all use the same style toolbar with the same primary menu options: Home, Insert, Page Layout.

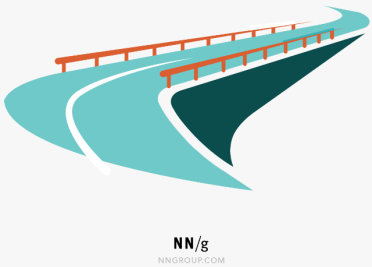
5. Error prevention

#5: Error prevention

Good error messages are important, but the best designs carefully prevent problems 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.

7:24

There are two types of errors: slips and mistakes. Slips are unconscious errors caused by inattention. Mistakes are conscious errors based on a mismatch between the user's mental model and the design.



Example of Usability Heuristic #5:
Guard rails on curvy mountain roads prevent drivers from falling off cliffs.

Tips

- Prioritize your effort: Prevent high-cost errors first, then little frustrations.
- Avoid slips by providing helpful constraints and good defaults.
- Prevent mistakes by removing memory burdens, supporting undo, and warning your users.

Learn more

- Full article: *Preventing User Errors*
- 3-minute video: *Error Prevention*

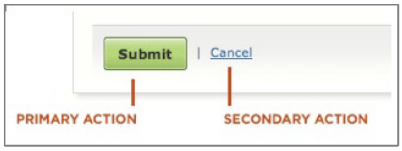
•Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
•Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Examples of error prevention design include:

- Requiring users to confirm a potentially harmful action before it is executed, such as deleting a file or sending an email
- Providing constraints to prevent users from entering incorrect data or making invalid selections
- Designing clear and concise error messages that guide users on how to fix the issue
- Providing clear feedback on the system's status to prevent users from performing actions at inappropriate times
- Anticipating and designing for potential user errors, such as using warnings or disabling certain actions in certain contexts.

Error prevention: examples

Yammer
Disables the update button after it is clicked, so the person cannot update the post twice by accident



Example from "Web form Design:Filling in the Blanks" by Luke W.
Make the primary action prominent with a larger click area. Cancel and other secondary actions are just shown as links

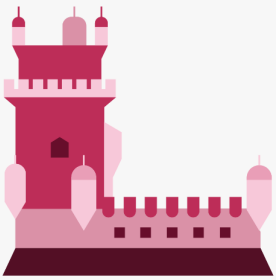
6. Recognition rather than recall

#6: Recognition rather than recall

Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.

7:24

Humans have limited short-term memories. Interfaces that promote recognition reduce the amount of cognitive effort required from users.



NN/g
NNGROUP.COM

Example of Usability Heuristic #6:
It's easier for most people to recognize the capitals of countries, instead of having to remember them. People are more likely to correctly answer the question Is Lisbon the capital of Portugal? *rather than* What's the capital of Portugal?

Tips

- Let people recognize information in the interface, rather than forcing them to remember ("recall") it.
- Offer help in context, instead of giving users a long tutorial to memorize.
- Reduce the information that users have to remember.

Learn more

- Full article: *Recognition vs. Recall in UX*
- 3-minute video: *Recognition vs. Recall*

●Minimize the user's memory load.

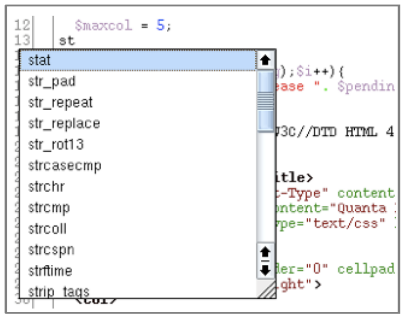
●Make objects, actions, and options visible.
●The user should not have to remember information from one part of the dialogue to another.
●Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Examples of Recognition rather than recall principle are:

- Providing menus, icons, and labels that are easy to recognize and understand
- Providing clear and concise instructions and help text within the interface

- Using familiar and consistent design patterns to minimize the user's cognitive load
- Providing feedback and confirmation messages that remind users of their previous actions and choices
- Using auto-complete or suggestion features to reduce the need for users to recall information
- Providing clear navigation paths and breadcrumbs to help users keep track of their location within the interface.

Recognition: examples



Quanta IDE
Auto completion for coding in a development environment

Keynote
Previews the fonts you can pick from, instead of just the font name

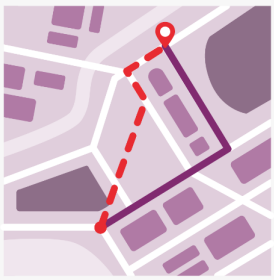
7. Flexibility and efficiency of use

#7: Flexibility and efficiency of use

Shortcuts — hidden from novice users — may speed up the interaction for the expert user so that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Flexible processes can be carried out in different ways, so that people can pick whichever method works for them.

7:24



Example of Usability Heuristic #7:
Regular routes are listed on maps, but locals with knowledge of the area can take shortcuts.

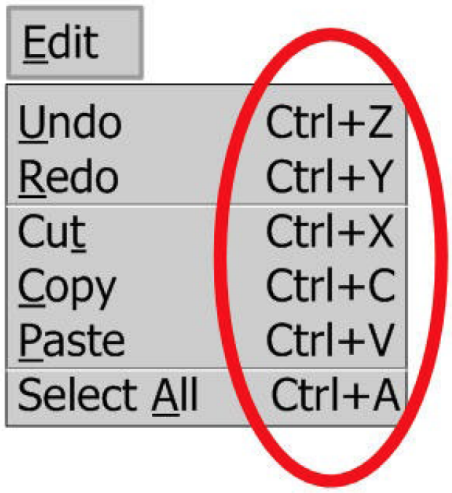
Tips

- Provide accelerators like keyboard shortcuts and touch gestures.
- Provide personalization by tailoring content and functionality for individual users.
- Allow for customization, so users can make selections about how they want the product to work.

Learn more

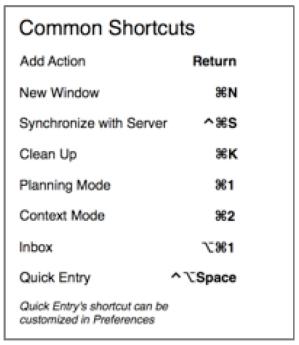
- Full article: *Flexibility and Efficiency of Use: The 7th Usability Heuristic Explained*
- 3-minute video: *Flexibility and Efficiency of Use*

-**Accelerators**—unseen by the novice user —may often speed up the interaction for the expert user so that the system can cater to both inexperienced and experienced users
-Allow users to tailor frequent actions and use shortcuts to save time and effort. Providing keyboard shortcuts, customizable toolbars or menus, and the ability to save user preferences can all enhance the efficiency and flexibility of the system.

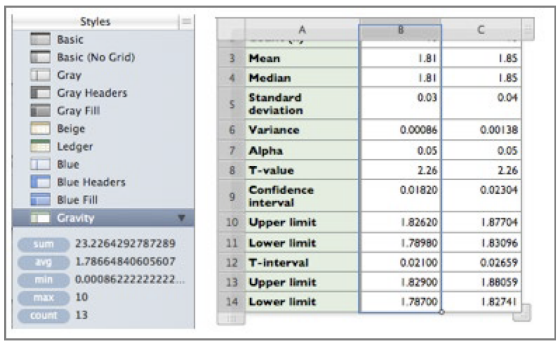


Flexibility and efficiency: examples

- Providing keyboard shortcuts for frequently used functions, which can save time and reduce the need to navigate menus
- Allowing users to customize the interface to their preferences, such as rearranging menus or changing the color scheme
- Providing options for different levels of complexity, so that expert users can perform tasks more quickly and efficiently than novice users
- Using autocomplete or suggestions to speed up data entry, such as predicting what a user is typing based on previous input or common patterns
- Providing context-sensitive help, so that users can quickly access information about a specific feature or function without having to search through documentation.



OmniFocus
List of keyboard shortcuts and accelerators



Numbers by Apple
Previews common function results on the left when a column is selected, more efficient than clicking on an action in the toolbar

8. Aesthetic and minimalist design

#8: Aesthetic and minimalist design

Interfaces should not contain information that is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.

This heuristic doesn't mean you have to use a flat design — it's about making sure you're keeping the content and visual design focused on the essentials. Ensure that the visual elements of the interface support the user's primary goals.

Example of Usability Heuristic #8:
An ornate teapot may have excessive decorative elements, like an uncomfortable handle or hard-to-wash nozzle, that can interfere with usability.

Tips

- Keep the content and visual design of UI focused on the essentials.
- Don't let unnecessary elements distract users from the information they really need.
- Prioritize the content and features to support primary goals.

Learn more

- Full article: *Aesthetic and Minimalist Design (Usability Heuristic #8)*
- 3-minute video: *Aesthetic and Minimalist Design*

●Dialogues should not contain information which is irrelevant or rarely needed

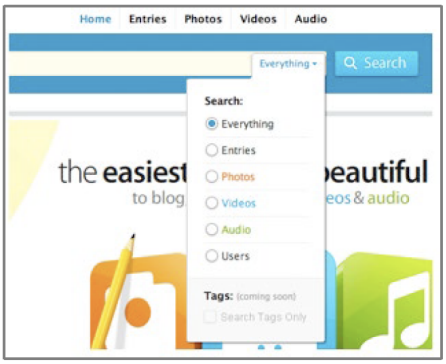
●Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility

●Visual layout should respect the principles of contrast, repetition, alignment, and proximity.

Examples of aesthetic and minimalist design include:

- Using a clean and simple interface design with minimal clutter
- Only including necessary information on each screen or page
- Using consistent design elements such as fonts, colors, and button styles
- Making sure that important information is visually highlighted and easy to find
- Using white space effectively to improve readability and overall visual appeal.

Aesthetics: example



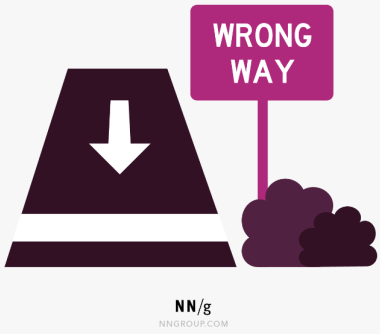
Kontain's search menu exemplifies the four principles of visual design:
1.Contrast: bold text is used for the two labels in the search
2.Repetition: the orange, blue, and green text match the media types
3.Alignment : strong left alignment of text, right aligned drop down
4.Proximity: a light rule is used to separate tags from the other options

9. Help users recognise, diagnose and recover from errors

#9: Help users recognize, diagnose, and recover from errors

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

These error messages should also be presented with visual treatments that will help users notice and recognize them.



Example of Usability Heuristic #9:
Wrong way *signs on the road remind drivers that they are heading in the wrong direction and ask them to stop.*

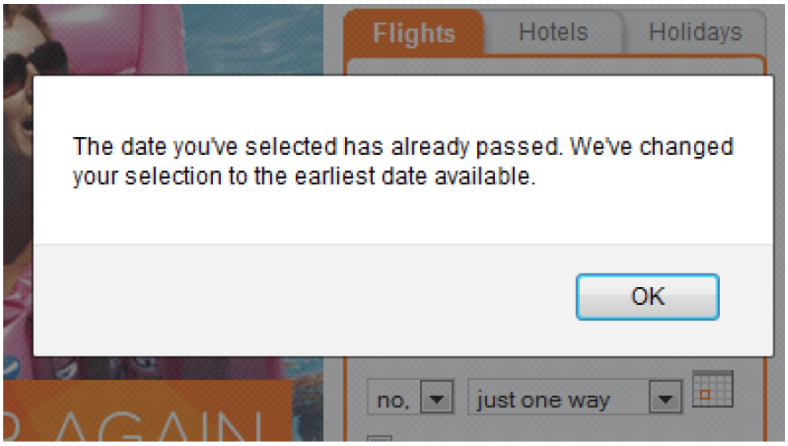
Tips

- Use traditional error-message visuals, like bold, red text.
- Tell users what went wrong in language they will understand — avoid technical jargon.
- Offer users a solution, like a shortcut that can solve the error immediately.

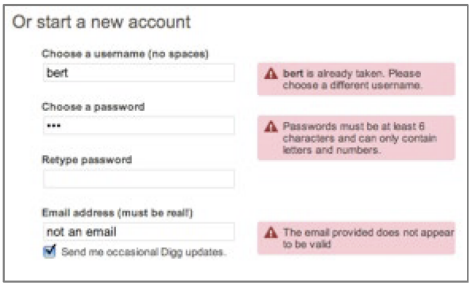
Learn more

- Full article: *Error-Message Guidelines*
- 2-minute video: *Helping Users Overcome Errors*

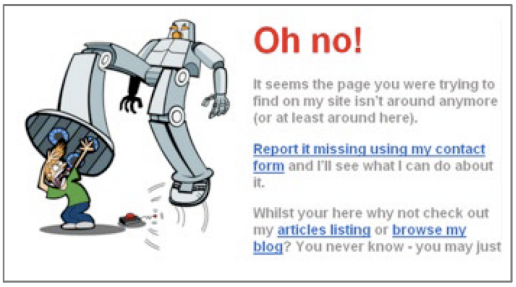
-Help users recognize, diagnose, and recover from errors.
-Error messages should be expressed in plain language (no jargon), precisely indicate the problem, and constructively suggest a solution.



Error recognition and recovery: examples



Digg
Provides immediate feedback with specific instructions



Humorous ' Page Not Found' Error
Uses a funny image and text, but provides viable alternatives (article listings and blog link) and a course of action (report it)

Some examples of how to help users recognize, diagnose, and recover from errors include:

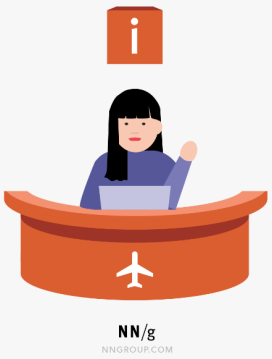
- Using clear and concise error messages that explain what went wrong and how to fix it.
- Providing suggestions or guidance on how to recover from the error.
- Using color or iconography to highlight errors and draw the user's attention to them.
- Using error prevention techniques, such as confirmation dialogs, to minimize the occurrence of errors.
- Allowing users to easily undo or redo actions, in case they make a mistake.

10. Help and documentation

#10: Help and documentation

It's best if the system doesn't need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

Help and documentation content should be easy to search and focused on the user's task. Keep it concise, and list concrete steps that need to be carried out.



Example of Usability Heuristic #10:
Information kiosks at airports are easily recognizable and solve customers' problems in context and immediately.

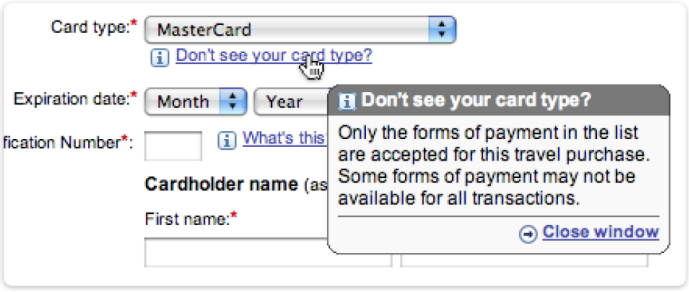
Tips

- Ensure that the help documentation is easy to search.
- Whenever possible, present the documentation in context right at the moment that the user requires it.
- List concrete steps to be carried out.

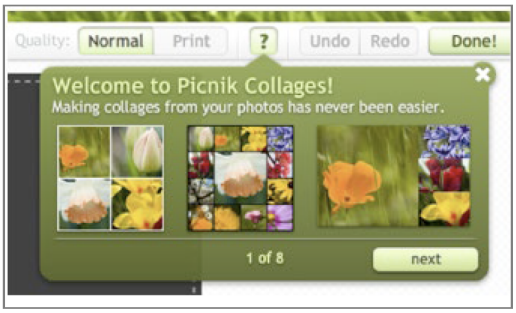
Learn more

- Full article: *Help and Documentation: The 10th Usability Heuristic*
- 3-minute video: *Help and Documentation*

-Even though it is better if software can be used without documentation, it may be necessary to provide help and documentation.
-Any such information should be contextual, easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.



Help and documentation: **examples**



Picnik
Contextual tips in Picnik are clear and easy to navigate

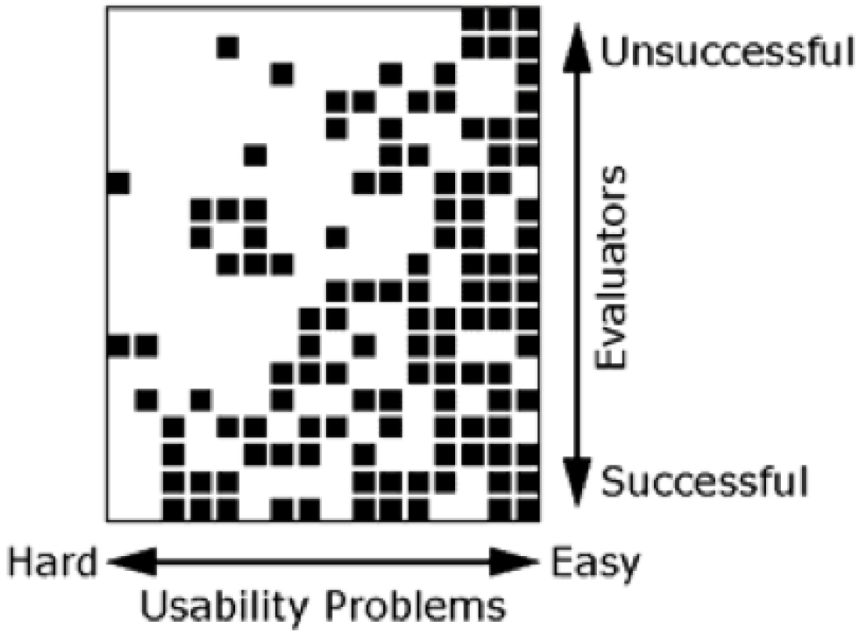
GoodBarry
Embedded videos can be used to showcase features as well as get people started using the product

Some examples of how to achieve this principle include:

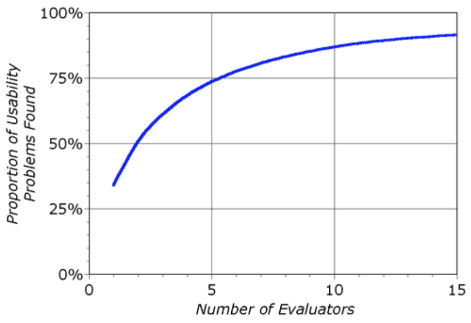
- Providing clear and concise instructions for completing tasks within the software.
- Offering tooltips or other forms of in-context help to explain unfamiliar features or concepts.
- Providing a searchable knowledge base or help center with detailed information on how to use the software.
- Including video tutorials or step-by-step guides to help users get started with the software.
- Offering a support chat or ticket system where users can get help from a support team if needed.

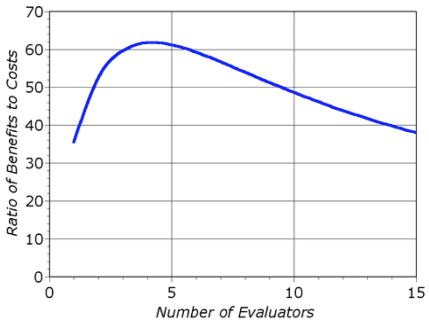
How many evaluators are needed for heuristic evaluation? (3-5)

According to Jakob Nielsen, a heuristic evaluation can be conducted with a minimum of three evaluators. However, he recommends having five evaluators as the optimal number to identify a majority of usability issues. Having more than five evaluators can start to yield diminishing returns in terms of finding new issues.



Practical considerations





Here are some practical considerations when conducting a heuristic evaluation:

1. Selecting evaluators: The evaluators should have expertise in user interface design and familiarity with the system being evaluated.
2. Conducting the evaluation: The evaluation can be done individually or in a group setting. It is important to provide clear instructions to the evaluators and ensure consistency in the evaluation process.

3. Reporting the findings: The findings should be presented in a clear and concise manner, highlighting the most critical issues and providing recommendations for improvement.
4. Follow-up: It is important to follow up on the recommendations and track the progress of any changes made to the system based on the evaluation findings.

How to run a heuristic evaluation

- Each of the 3 –5 evaluators does a heuristic evaluation of an interface alone.
- Sometimes an observer can record the evaluator's comments, sometimes the evaluator does it.
- Observers **can** answer evaluators' questions, in contrast to traditional user testing, particularly if it's not a walk up and use system.
- Heuristic evaluation can be done on paper prototypes.
- Heuristic evaluations typically last 1 –2 hours, but it does depend on the complexity of the software.
- The expert goes through the interface several times –first time to get a feel for the system, second time to focus on specific elements
- Evaluators can be given scenarios that describe typical usage scenarios (built from a task analysis of users)
- Evaluators produce a list of usability problems: the usability principle and the design feature that violated it

Benefits of heuristic evaluation

1. Cheap
2. Relatively easy
3. Instant gratification lists of problems are available immediately after the inspection
4. It can be carried out with low numbers of participants
5. Fits in with most software development processes
6. Cost effective

Drawbacks of heuristic evaluation

1. Important issues may get missed
2. Might identify falseissues
3. Many trivial 不重要的 issues are often identified, making it seem overly critical
4. Experts have biases

Some quizes:

1. Which of the following is not a benefit of Think Aloud evaluation?
   a) Provides insights into the user experience
   b) Identifies issues with the software
   c) Can be used with a finished product
   d) Inexpensive and easy to carry out
   e)none of the above

2. Heuristic evaluation is also known as:
   a) User testing
   b) Analytical evaluation
   c) Empirical evaluation
   d) Expert evaluation

3. How many evaluators are needed for a heuristic evaluation?
   a) 1-2
   b) 3-5
   c) 6-8
   d) 9-10

4. Which heuristic principle is concerned with minimizing the user's memory load?
   a) Visibility of system status
   b) Match between system and real world
   c) Flexibility and efficiency of use
   d) Recognition rather than recall

5. Which heuristic principle suggests that every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative

visibility?
a) User control and freedom
b) Aesthetic and minimalist design
c) Help users recognize, diagnose, and recover from errors
d) Match between system and real world

6. What is the benefit of involving observers in a heuristic evaluation?
   a) They can answer evaluators' questions
   b) They provide a second opinion on the evaluators' findings
   c) They can perform the evaluation on behalf of the evaluators
   d) They can provide insights into the user experience

1. e
2. d
3. b
4. d
5. b
6. a

More quiz:

1. Post-test questionnaires (conducted after a usability test) are particularly useful for measuring
   a) safety.
   b) efficiency.
   c) learnability.
   d) user satisfaction
2. A pluralistic walkthrough
   a) is usually conducted at the end of the development process.
   b) is often conducted with low-fidelity(lo-fi) designs.

c) requires having several alternate designs.
d) requires a fully functional prototype.

3. Providing accelerators (e.g. keyboard shortcuts) mostly addresses
   a) utility.
   b) efficiency.
   c) learnability.
   d) user satisfaction.
4. A method that does not require human participants serving as test users is the
   a) usability test.
   b) pluralistic walkthrough.
   c) Rubin and Chisnel’s comparison test.
   d) heuristic evaluation.
5. The cognitive walkthrough mainly evaluates a product's
   a) utility.
   b) efficiency.
   c) learnability.
   d) likeability.

1. d
2. b
3. b
4. d
5. c

1) ----------is as an attempt to introduce psychological theory into the informal and subjective walkthrough technique
A. heuristic evaluation
B. model based evaluation
C. cognitive walkthrough
D. review based

2) What are the things needed for cognitive walkthrough?
A. prototype of the system
B. description of the task
C. both a and b
D. none of the above

3) Which of the following is golden rule for interface design?
A. strive for consistency
B. offer error prevention and simple error handling
C. reduce short-term memory load
D. all of the mentioned

4) Which of the following are not the components of the HCI approach to design?
A. Tasks
B. Humans
C. Usability
D. Technology

5) Building things from user's perspective is called ___
A. Functionality
B. Usability
C. Portability
D. None

1. c
2. c
3. d
4. b
5. b

7b-Qualitative Evaluation