10个UI设计的经验性法则（尼尔森）

1.清晰可见的系统状态
通过准确而合理的反馈，用户应该始终知道系统当前的状态

2.系统和现实世界保持一致
系统应该使用用户的语言，包括词汇、短语以及用户所熟悉的概念，而不是以系统为导向的术语。并且遵循现实世界的约定俗成，用自然而符合逻辑的顺序来呈现信息。

3.用户操控的自由度
用户常常会不经意间选（用）错系统的功能，因此他们需要一个被清楚标出的“紧急出口”来迅速离开当前的错误场景，而不是必须先经历一系列冗余的对话框。同时，应该提供撤消和重做的操作。

4.一致性和标准化
不要用不同的词汇、场景或者行动点来表示同一件事物，用户会因此而感到困惑。尽量遵循平台化的通用约定或标准。

5.错误的预防
一个优秀的设计可以在第一时间预防可能发生的错误，这比单纯提供错误反馈信息要来的好。同时，在用户实施行动之前，提供一个确认选项，尽可能的检查并消除错误发生的可能。

6.再识别比短期回忆好
尽量把重要的元素、行动点以及选项可视化从而减轻用户的记忆负担。不应该强迫用户记住一个对话场景中不同部分的信息。系统的介绍（说明）信息应该清晰可见，或在合适的时候可以被轻松地检索到。

7.灵活性和高效性
加速器（高级功能）— 对新手用户来说应是不可见的，但是又常常可以提高专家用户和系统的交互效率：这样一来就可以同时满足无经验和有经验的用户。允许用户自定义常用的操作。

8.美学和最简设计
对话场景不应该含有无关或不必要的信息。每个多余的无关元素都会对必要元素产生削弱并降低它们的可视度。

9.帮助用户识别、诊断并解决错误
出错信息应该用浅显易懂的语言表达出来（而不是代码语言），不仅准确地描述问题本身，还应该提供有建设性的解决方案。

10.帮助和说明文档
尽管一个不需要说明文档就能使用的系统也许是最好的，但还是有必要提供一份帮助和说明文档。文档里的任何信息都应该能被轻松的检索到。这些信息简洁明了，专注于说明用户的任务，并列出需要实施的具体步骤。

10 Usability Heuristics for User Interface Design

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 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.

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.