



人机交互的软件工程方法 —— 简易可用性工程

主讲教师：冯桂焕

fgh@software.nju.edu.cn

2012年春季



背景概述



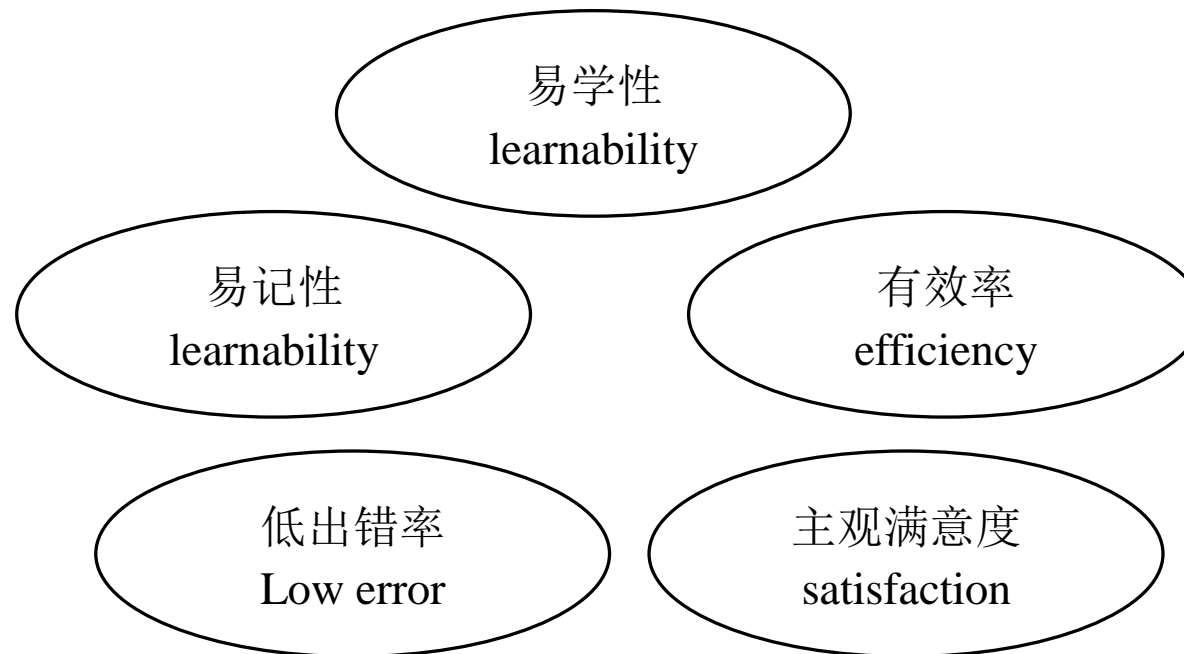
- 软件产品的用户群体已发生巨大转变
 - 以往：热爱技术的专业人员
 - 现在：缺乏耐心的消费者
- 用户不再是麻烦
 - “用户友好”的软件？
 - 用户希望在完成任务的时候，机器不要碍手碍脚
 - 不同用户的需求各异，不能从系统单方面友好
 - “可用”的软件
 - 使产品易学易用等
 - “用户体验”
 - 使用户喜欢产品



可用性目标



- 可用性目标不仅涉及人与之正在发生交互作用的系统，还包括系统对使用它的人所产生的作用





易学性 (learnability)

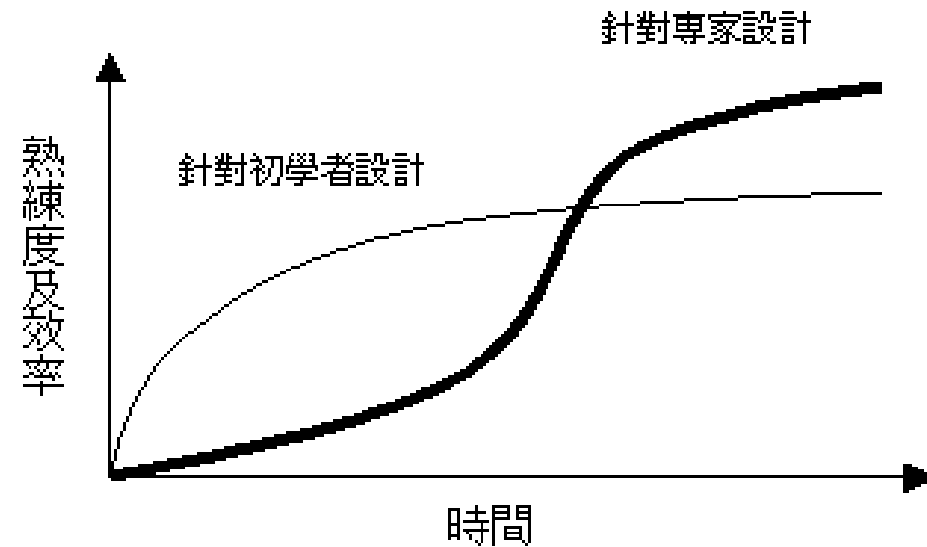


■ 解释

- 指使用系统的难易，即系统应当容易学习，从而用户可以在较短时间内应用系统来完成某些任务
- 最基本的可用性属性

■ 易学性对应系统学习曲线的开头部分

- “10分钟法则”
?



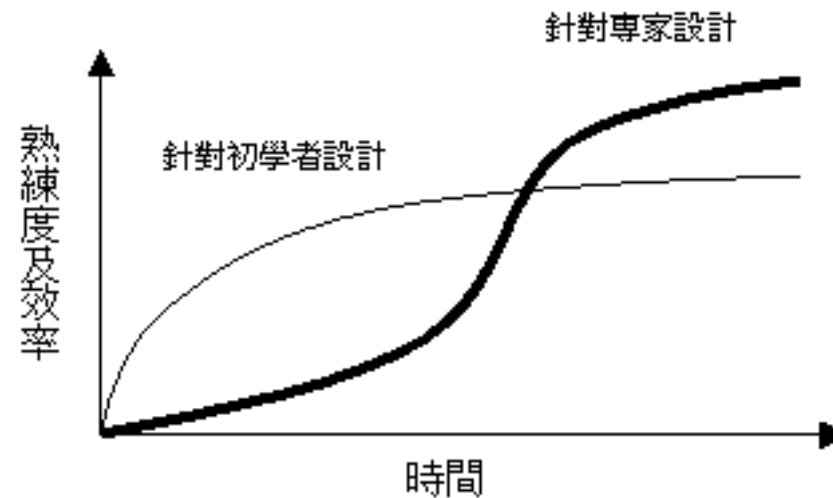


有效率 (efficiency)



■ 解释

- 当用户学会使用产品之后，用户应该具有更高的生产力水平（效率）
- 效率指熟练用户到达学习曲线上平坦阶段时的稳定绩效水平





易记性 (memorability)



■ 解释

- 用户在学会使用软件后应当容易记忆
- 学会某个系统后，应能够迅速回想起它的使用方法

■ 易记性的影响因素

- 位置：将特定对象放在固定位置
- 分组：对事物按照逻辑进行恰当的分组
- 惯例：尽可能使用通用的对象或符号
- 冗余：使用多个感知通道对信息进行编码

■ 启发

- 良好组织，使用用户已有的经验帮助提高易记性



低出错率（errors）



■ 解释

- 人是会犯错误的
 - 有些错误会被用户发现并纠正
 - 有些错误会带来灾难性后果

■ 措施

- 保证导致灾难性后果错误的发生频率降到最低
- 保证错误发生后迅速恢复到正常状态



主观满意度 (satisfaction)



■ 解释

- 用户对系统的主观喜爱程度
- 某些情况下，系统的娱乐价值比完成任务的速度更为重要
 - 如家用计算、游戏等非工作环境的系统

■ 观念的转变

- 传统软件质量观
 - 侧重内部效率和可靠性
 - 如程序代码运行时的效率以及灵活性、可维护性
- 人机交互软件质量观
 - 转向用户视角



用户体验目标



■ 问题

- 随着新技术渗透到人们的日常生活中，人们对产品有了更多的要求
- 到底什么样的产品才是用户愿意使用和购买的？
 - 让用户感到满意并留下愉快主观感受的产品更可能被多次使用

■ 什么是用户体验

- 用户在与系统交互时的感觉
- 为儿童创建的网站应该要有趣并且引人入胜
- 面向年轻人的网站则应该更注重时尚感和趣味性
- 较可用性目标主观
- 可用性可能对用户体验带来阻碍



简易可用性工程



■ 特点

- 以提高产品的可用性为目标的先进的产品开发方法论
- 借鉴了许多不同领域的方法和技术
- 强调以人为中心来进行交互式产品的设计研发

■ 历史

- 上世纪80年代获得工业应用
- 90年代得到迅速普及

■ 实例

- IBM公司：“可用性方面的投入是一本万利的”
- MS已有14个可用性实验室近200名可用性专业员工



可用性度量



■ 常用方法

- 选择一些能够代表目标用户群体的测试用户
- 让这些用户使用系统执行一组预定的任务
- 比较任务的执行情况
- 针对多维属性
 - 取每个可用性属性的平均值
 - 查看整体分布情况

■ 主观满意度度量举例

- 在1~5分的5分制情况下平均值至少为4
- 或至少50%的用户给系统打5分
 - 或给系统打1分的用户不超过5%



注意事项



- 度量一定要针对特定的用户和特定的任务进行
- 用户对不同任务的可用性结果预期可能不同
 - 用于编辑邮件的文字处理程序和用于编写数万页技术文档的文字处理程序的要求是不同的
- 因此测试前要明确一组具有代表性的测试任务



易学性度量



■ 可用性属性中最容易度量的属性

- 找一些从未使用过系统的用户
 - 能够代表系统的目标用户
 - 区分没有任何计算机使用经验的新手用户和具有一般计算机使用经验的用户
- 统计他们学习使用系统直至达到某种熟练程度的时间

■ 特定熟练程度

- 用户能够完成某个特定的任务
- 或用户能够在特定的时间内完成一组特定任务
- 原因：学习曲线没有明确区分“学会和未学会”



使用效率度量



- 并不是所有用户都能够迅速达到最终的绩效水平
 - 用户自身的原因
 - 少量系统的操作十分复杂
- 同样要区分不同的用户群体
 - 对于有经验的用户
 - “有经验”较为正规的衡量方式是通过使用系统的小时数来定义的
 - 先使用，然后度量其绩效水平
 - 或为用户绘制学习曲线
 - 当发现用户的绩效水平在一段时间内不再提高时，就认为已经达到了该用户的稳定绩效水平



易记性度量



■ 用户分类

- 新手用户，熟练用户，非频繁使用用户
- 对非频繁使用用户进行测试最能体现系统的易记性

■ 度量方法

- 对在特定长时间内没有使用系统的用户进行标准用户测试
 - 记录下这些用户执行特定任务所用的时间
- 对用户进行记忆测试
 - 如在用户完成一个应用系统的特定任务后，让用户解释各种命令的作用



错误率度量



- 错误
 - 通常指不能实现预定目标的操作
- 度量
 - 在用户执行特定任务时通过统计这种操作的次数
 - 可以在度量其他可用性属性的同时来度量
- 错误分类
 - 错误发生后能够被用户立刻纠正，不会对系统带来灾难性的影响
 - 往往会被包含在使用效率的统计当中
 - 不易于被用户发现，从而可能造成最终结果存在问题
 - 设计人员在设计时也应该将其发生的频率降到最低



满意度度量



- 满意度度量评价都是主观的
 - 以询问用户的方式进行度量更合适
 - 为减少单个用户评价的主观性，把多个用户的结果综合起来取其平均值
- 度量通常在用户测试完成后进行
 - 要求用户通过简单的调查问卷对系统打分
 - 可以1-5或1-7的Likert度量尺度或语义差异尺度作为打分标准
 - 一定要在用户使用系统执行真实的任务之后再来询问他们的看法



调查问卷的设计



- 通常设计得较为简短
 - 以保证最高的结果返回率
- 以1-5或1-7的Likert度量尺度或语义差异尺度作为打分标准
 - 得分越高，说明认可的程度越高
 - 研究发现1-5分的评价尺度的中值是3.6分（1分满意度最低，5分最高）
 - 评估定义一个锚点或基准点是非常重要的
- 不论采用什么样的评价尺度，都应当在大规模测试前进行试点测试



Likert度量尺度举例



- 对于下面关于**系统**的陈述，请指出您同意或不同意的程度：
 - “很容易学会怎样使用这个系统”
不满意 1 2 3 4 5 满意
 - “使用这个系统是一段让人很沮丧的经历”
 - “这个系统可以帮助达到很高的生产效率”
 - “担心使用该系统获得的结果存在错误”
 - “使用该系统工作让人感觉很愉快”



语义差异尺度标准



- 请在最能够体现您对这个系统印象的位置上做标记
- 愉快 — — — — — 气恼
- 完善 — — — — — 不完善
- 合作 — — — — — 不合作
- 简单 — — — — — 复杂
- 快速 — — — — — 慢速
- 安全 — — — — — 不安全



图标的可用性度量举例



- 不是所有图标都有好的可用性特征
- 如何度量？
 - 对每一个可用性属性，定义出可度量的标准
 - 前提：弄清图标出现的环境及使用场合



度量方法一：经典方法



- 为系统设计四套不同图标，每套17个
 - 测试每一个图标的易学习性、使用效率和主观满意度
- 易学习性
 - 展示一个图标，问“你认为这是什么意思”（测试直觉性）
 - 展示一套图标，测试可理解性
 - 讲出一个图标的名字及功能的简短描述，让用户指出匹配的图标
 - 及给出一套图标的名字，让用户指出相应匹配
 - 得分：被正确描述或命名的图标所占的比例



■ 效率测试

- 方法一：学习+给出图标名字+随机显示一个+用户选择
- 方法二：学习+给出图标名字+随机显示若干+用户选择
- 得分：用户的反应时间（秒）

■ 主观满意度

- 方法一：就图标是否容易识别打分
- 方法二：给出一个概念，让用户从四个可能图标中选择
- 得分
 - 方法一：给图标的打分
 - 方法二：选择正确图标的用户比例



四种主要技术



- 完整的可用性工程过程
 - 了解用户
 - 竞争性分析
 - 设定可用性目标
 - 用户参与的设计
 - 迭代设计
 - 产品发布后的工作
- 简化
 - 用户和任务观察
 - 场景（**scenario**）
 - 简化的边做边说（**thinking aloud**）
 - 启发式评估



■ 用户和任务观察

- 了解产品的目标用户是可用性工程的第一个步骤
- 注意
 - 要直接与潜在用户进行接触
 - 不要满足于间接的接触和道听途说

■ 场景

- 简便易行的原型工具
- 通过省略整个系统的若干部分来减少实现的复杂性
- 水平原型：减少功能的深度并获得界面的表层
- 垂直原型：减少功能的数量而对所选功能进行完整实现
- 可以是纸质模型，也可以是简单的**RAD**原型



■ 边做边说法

- 让真实用户在使用系统执行一组特定任务的时候，讲出他们的所思所想
- 最有价值的单个可用性工程方法
- 可了解用户为什么这样做，并确定其可能对系统产生的误解
- 实验人员需要不断地提示用户，或请他们事先观摩

■ 启发式评估

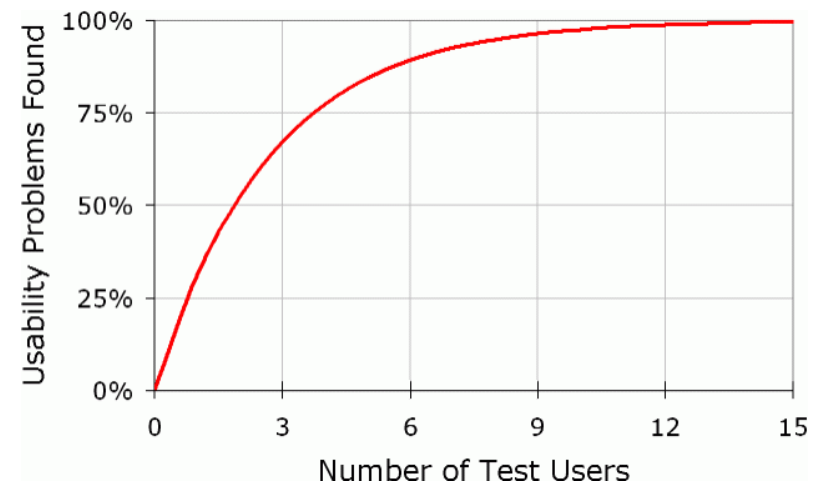
- 研究表明，能够发现许多可用性问题
 - 剩下的可以通过简化的边做边说方法来发现
- 为避免个人的偏见，应当让多个不同的人来进行经验性评估



启发式评估



- 问题：究竟需要多少个测试专家参与
- n 个测试专家能够发现的可用性问题数量
 - $N(1-(1-L)^n)$
 - N : 设计中存在的可用性问题的总数
 - L : 单个参与者所能够发现的可用性问题的比例（经验取值约为31%）
- 结论
 - 5名专家能够发现约80%的可用性问题
 - 被认为是最恰当的可用性测试用户数量
 - 建议将测试分阶段进行





设计规则



■ 说明

- 这些规则大多来源于提出者的经验和总结
- 不是完美无缺的，甚至有些会相互矛盾
- 在具体使用时，必须根据实际情况进行调整和细化

■ 基本规则，by Alan Dix

- 可学习性
 - 新用户能用它开始有效的交互并能获得最大的性能
- 灵活性
 - 用户和系统能以多种方式交换信息
- 健壮性
 - 在决定成就和目标评估方面对用户提供的支持程度



可学习性



原理	定义	举例
可预见性	用户能够基于以往的交互经验确定系统可能的行为	图形工具包中含有许多图形对象（矩形、圆形等）。用户下一次使用的时候能够确定图形是由哪些图形对象构成的
同步性	支持用户在当前状态下评估过去操作的结果	系统应该对用户操作给出显式反馈，如用户执行文件复制或移动操作后，应该在目标文件夹显示一个新的文件名等
熟悉性	与新系统交互时，能应用其他系统或领域的知识的程度	隐喻是该原则的例子，如“桌面”将文件、文件夹等真实桌面任务相关的概念应用的计算机当中。屏幕上物体的形状表示了其功能。
普遍性	用户能够从同一领域或跨领域应用的交互中使交互知识得到扩展	同一应用中，用户能够应用绘制矩形的经验绘制一个圆。另外不同应用之间的复制、剪切、粘贴命令是一个很好的跨领域普遍性的例子
一致性	能够从相似任务或情况中得出输入输出行为的相似性	同一系统中不同命令参数使用的一致性



灵活性



原理	定义	举例
能动性	原则上建议给用户更大的主动权，并减少系统的主动权	用户能够在交互过程的任意时刻开始或终止某个操作
多线程	允许用户同时执行多个任务的能力	窗口系统中，用户可以在一个窗口进行文本编辑，同时在另一个窗口进行文件管理
任务可移植性	用户与系统之间进行控制转移的能力	文档拼写检查既可以由系统自动完成，也可以由用户或二者合作完成
可替换性	相等的输入或输出之间可以相互替换的能力	文档的页边距设置既可以以英寸为单位，也可以以厘米为单位
可定制性	用户或系统对界面的可修改能力	如用户可对 MS-Word 的工具条进行定制，以保证常用功能选项总是可见的



健壮性



原理	定义	举例
可观察性	用户能够在多大程度上根据系统的表现推测系统的状态	从 ftp 上下载文件的时候会显示一个进度条表明下载进度。如果进度条从界面上消失，表明下载完成
可恢复性	当用户行为导致系统错误的时候，提供给用户执行正确操作的支持	MS-Word 中“undo”和“redo”功能能够帮助用户恢复到前一个或者后一个状态
反应性	用户能够在多大程度上预测系统的响应时间	任何系统中加载程序的时候都需要占用一定时间
任务一致性	系统提供给用户的帮助与用户要执行任务的一致性，以及任务自身与用户理解之间的一致性	系统需要提供用户必须的服务，实现方面要满足用户对服务的理解



黄金规则



- 1. 尽可能保证一致
- 2. 符合普遍可用性
- 3. 提供信息丰富的反馈
- 4. 设计说明对话框以生成结束信息
- 5. 预防并处理错误
- 6. 让操作容易撤销
- 7. 支持内部控制点
- 8. 减轻短时记忆负担



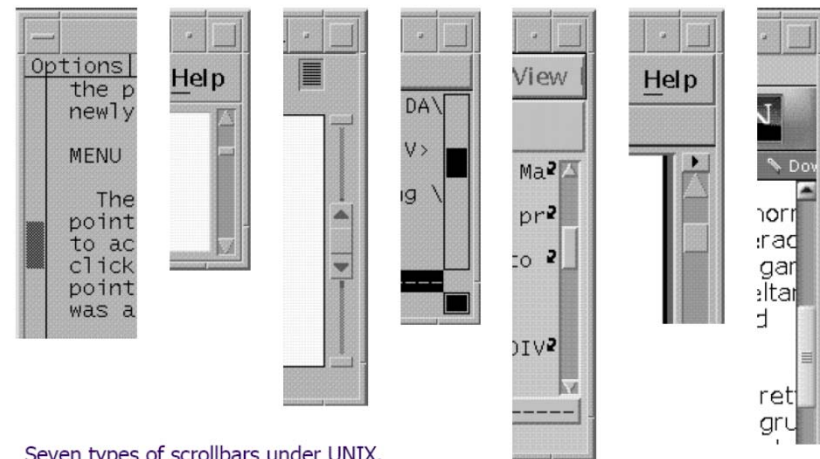
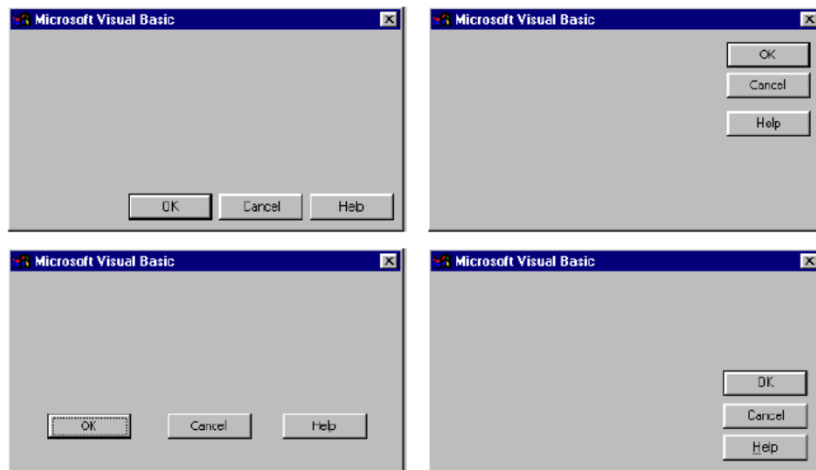
Ben Shneiderman



1. 尽可能保证一致



- 一致性让界面变得熟悉和可预测
- 最容易被违背的原则
 - 相似操作下一致的动作序列
 - 菜单、帮助中一致的术语
 - 一致的颜色、布局、字体等



Seven types of scrollbars under UNIX.



2. 符合普遍可用性



- 充分考虑用户操作的熟练程度、年龄范围、身体状况（如是否有残疾）等多方面的不同需求
- 专家用户
 - 缩写或快捷键操作，以丰富界面可感知的系统质量
- 新手用户
 - 尽可能提供引导性的帮助信息，帮助用户完成特定的交互任务

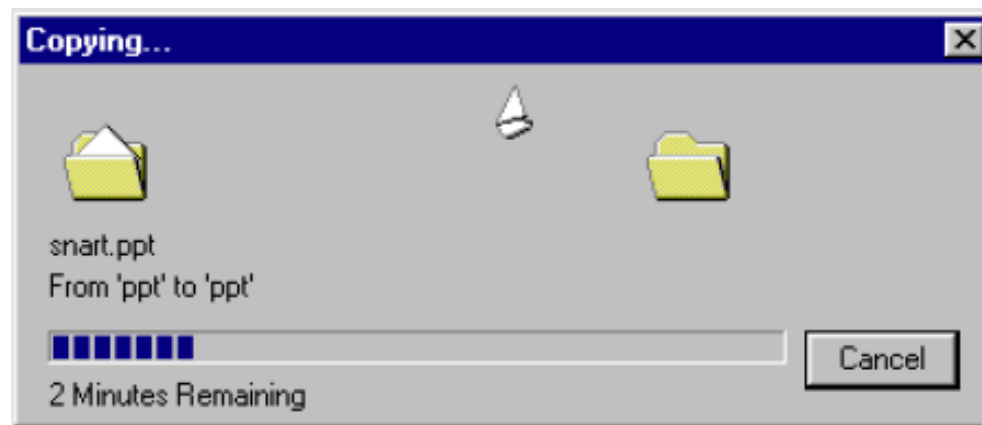
Edit View Insert For	
Undo Typing	⌘Z
Repeat Typing	⌘Y
Cut	⌘X
Copy	⌘C
Paste	⌘V
Paste Special...	
Paste as Hyperlink	
Clear	
Select All	⌘A
Duplicate	⌘D
Delete Slide	
Find...	⌘F
Replace...	⌘H
Go to Property...	
Object	



3. 提供信息丰富的反馈



- 要求
 - 对常用操作，则反馈信息可以相对简短
 - 对不常用操作，系统的反馈信息就应该丰富一些
- 途径：界面对象的可视化表现

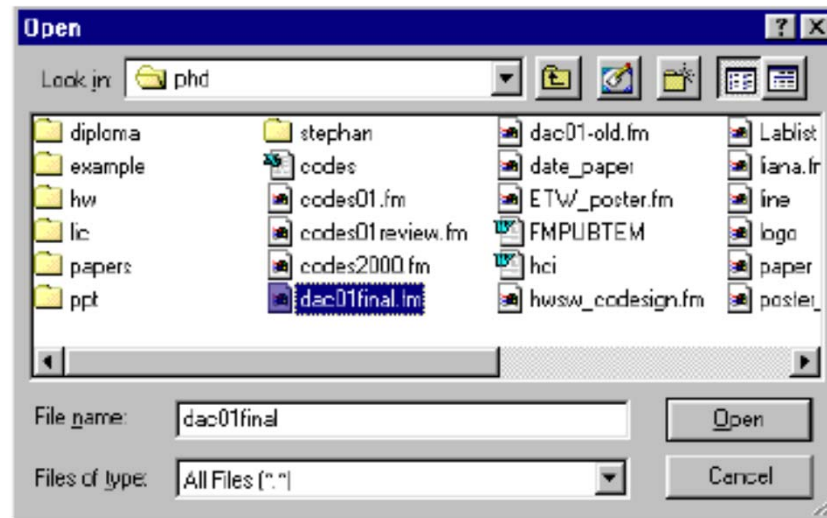




4. 设计说明对话框以生成结束信息



- 目的：让用户知道什么时候他们已经完成了任务
- 作用
 - 使用户产生完成任务的满足感和轻松感
 - 有助于让用户放弃临时的计划和想法

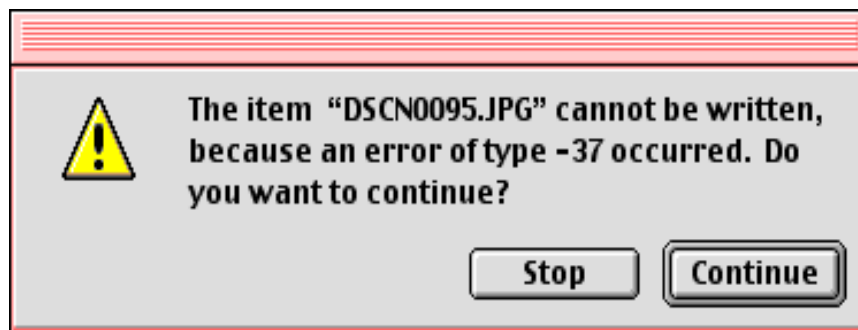




5. 预防并处理错误



- 目的：提供故障预防和简单的故障处理措施
- 作用：用户错误能够在清晰的指导下进行恢复
- 错误预防
 - 将不适当的菜单选项功能以灰色显示屏蔽
 - 禁止在数值输入域中出现字母字符
- 错误处理
 - 提供简单的、有建设性的、具体的指导来帮助用户恢复操作

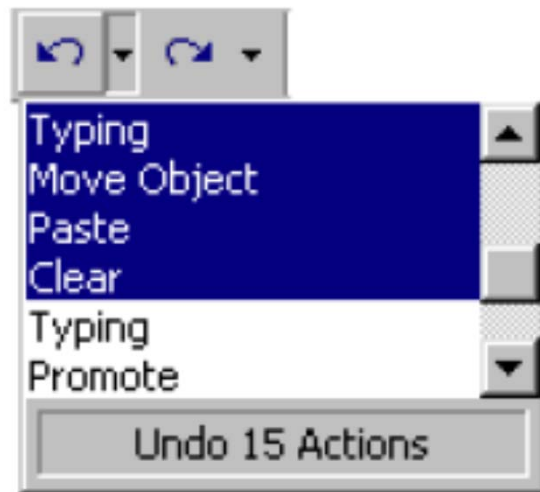




6. 让操作容易撤销



- 目的：减轻用户的焦虑情绪，并鼓励用户尝试新的选项
 - 可以是单独的操作
 - 也可以是一个数据输入任务或一组完整的操作等

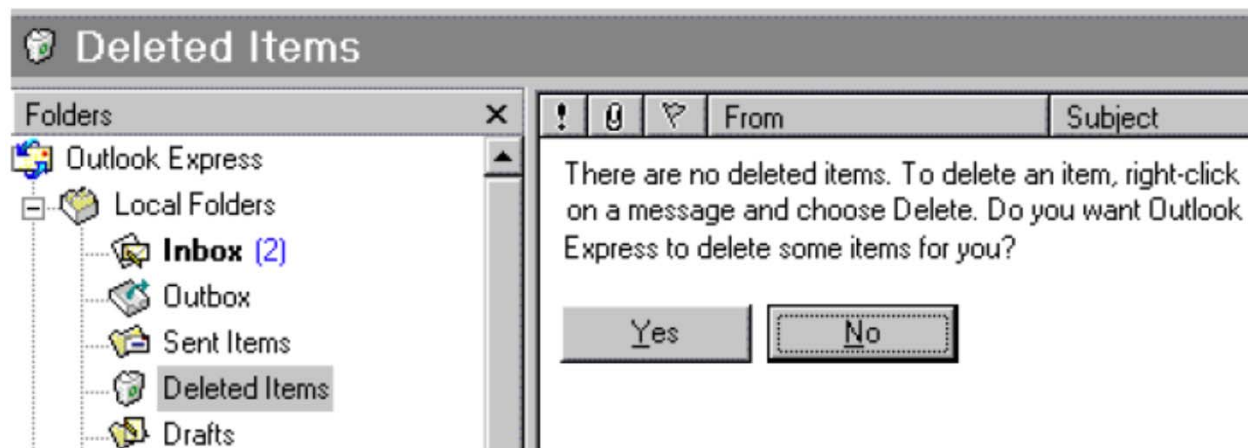




7. 支持内部控制点



- 鼓励用户成为行为的主动者而不是响应者
- 措施
 - 避免模态对话框
 - 避免很长的引导序列
 - 提供出口：取消、重做、放弃等





8. 减轻短时记忆负担



- 出发点：人凭借短时记忆存储的信息是非常有限
- 措施
 - 界面显示尽可能简单
 - 不同显示页面的风格应该统一
 - 尽可能减少在窗口之间的移动
 - 并且要确保提供用户足够的学习代码、记忆操作方法和操作序列的时间
 - 提供适当的在线帮助信息



十项启发式规则



- 系统状态的可见度
- 系统和现实世界的吻合
- 用户享有控制权和自主权
- 一致性和标准化
- 避免出错
- 依赖识别而非记忆
- 使用的灵活性和高效性
- 审美感和最小化设计
- 帮助用户识别、诊断和恢复错误
- 帮助和文档



Jacob Nielsen

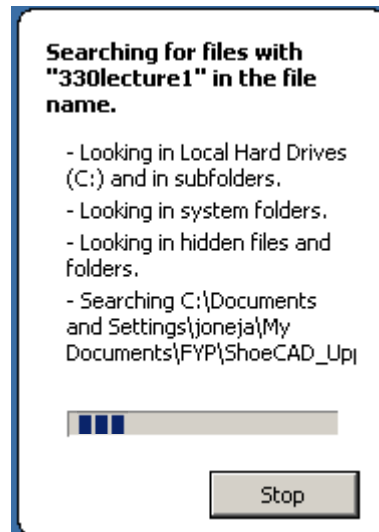


1. 系统状态的可见度

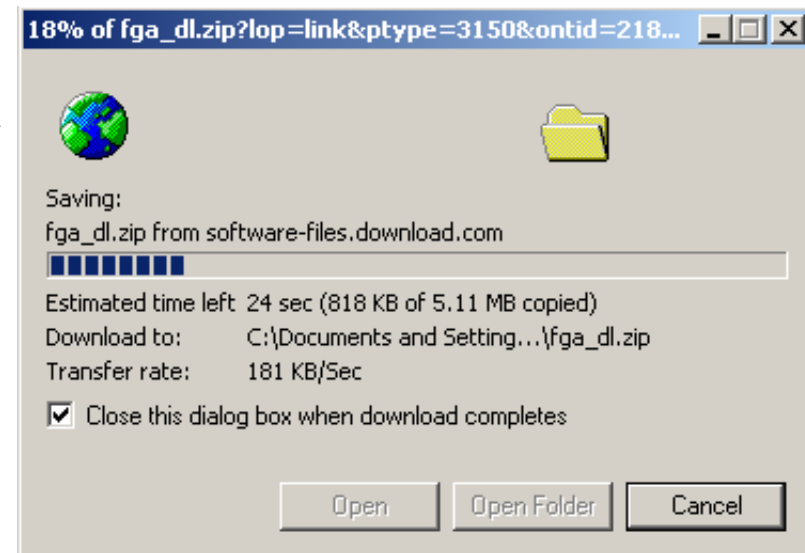


- For any activity expected to take over 3-5sec, give status feedback to user

MS Windows: search
No estimate of time



IE: download
Better feedback



- For each action, system should respond in some way; e.g. in a web form, clicking a submit button → button changes color, or a clicking sound is made



2.系统和现实世界的吻合

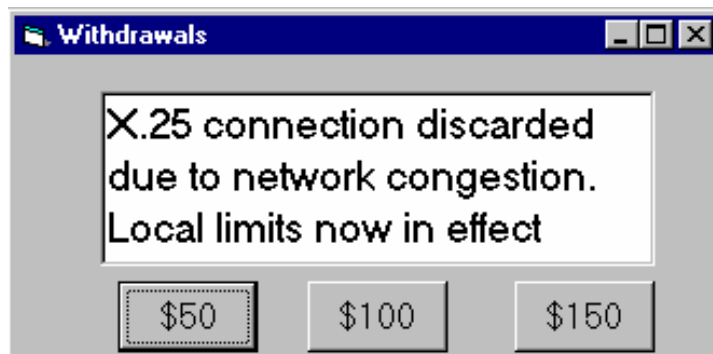


Example:

ATM machine message when trying to withdraw some money:

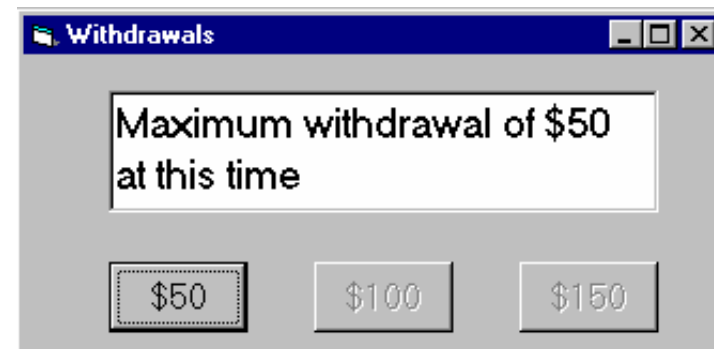
Poor:

User does not care what is X.25
What is the 'Local limit' ?



Better:

Tells user what they can do;
Blocks out restricted actions





3. 用户享有控制权和自主权

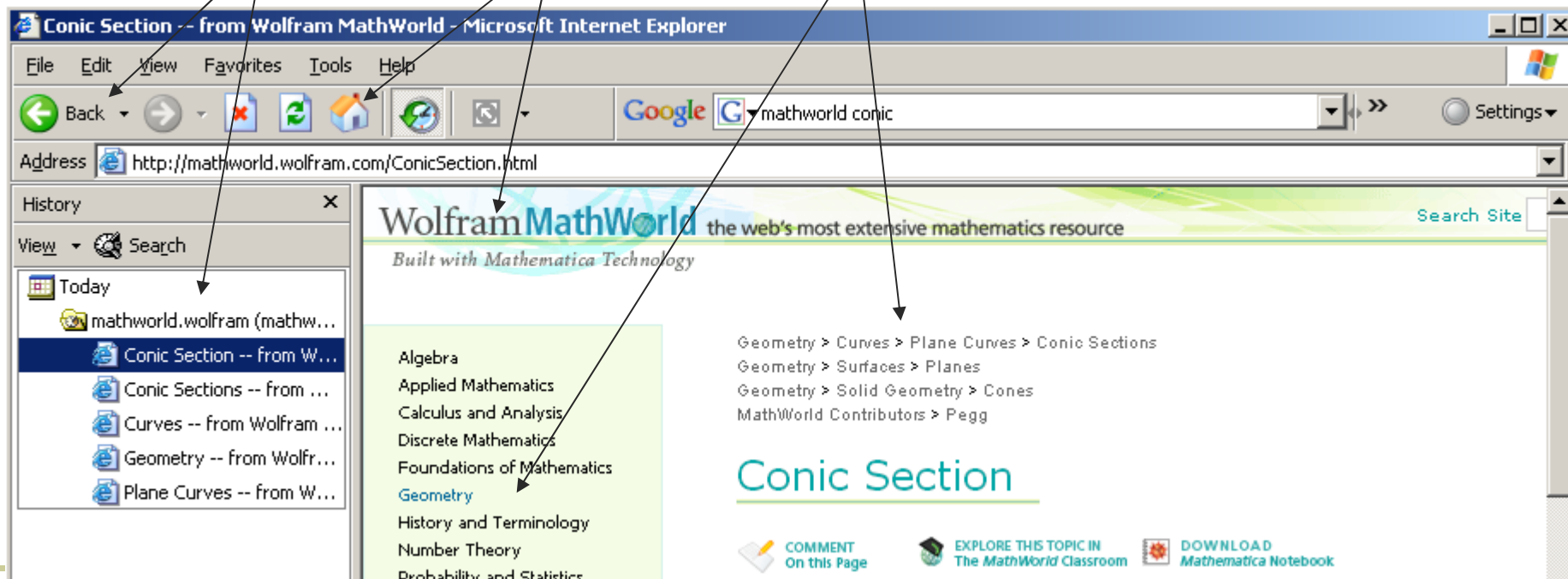


- For choice made by error, provide a ‘way back’ [e.g. undo, redo], or a method to re-start [e.g. Home button on website]

undo, redo

link to home

where you are





4.一致性和标准化



- Do not use multiple words/names for same function in different places
- Consistent terminology in prompts, menus, and user-guides
- Use icons/images without ambiguous meanings
- Consistent color, layout, capitalization, and fonts throughout the application

MS Internet explorer:
Search in page, or Print Preview?



US road signs:
Which one is for “curvy road ahead?”





5.避免出错



Avoid possibility for user to make errors

No calendar !

The screenshot shows the HKUST Booking System interface. At the top, there is a banner with the text "HKUST BOOKING SYSTEM for Sports Facilities and Student Amenities". Below the banner is a navigation bar with icons and labels: Main, Facility Info, Facility General Booking, My Record, FAQ, and Logout. The user is logged in as "User : [JONEJA Ajay]" and the date is "Today: 02/09/2007". The main heading is "General Booking - For Multi-facilities on a day". Below this, there is a form with the following fields:

- Required Date: (DD/MM/YYYY)
- Required Facility: select from the list below, max. 7 at a time

There is a "Submit" button to the right of the date field. Below the form, there are three columns of facilities, each with a checkbox:

AMENITIES - LG4	AMENITIES - Seafront	SPORTS - LG1
<input type="checkbox"/> Band Room	<input type="checkbox"/> BBQ Stove 1	<input type="checkbox"/> Badminton Court 1



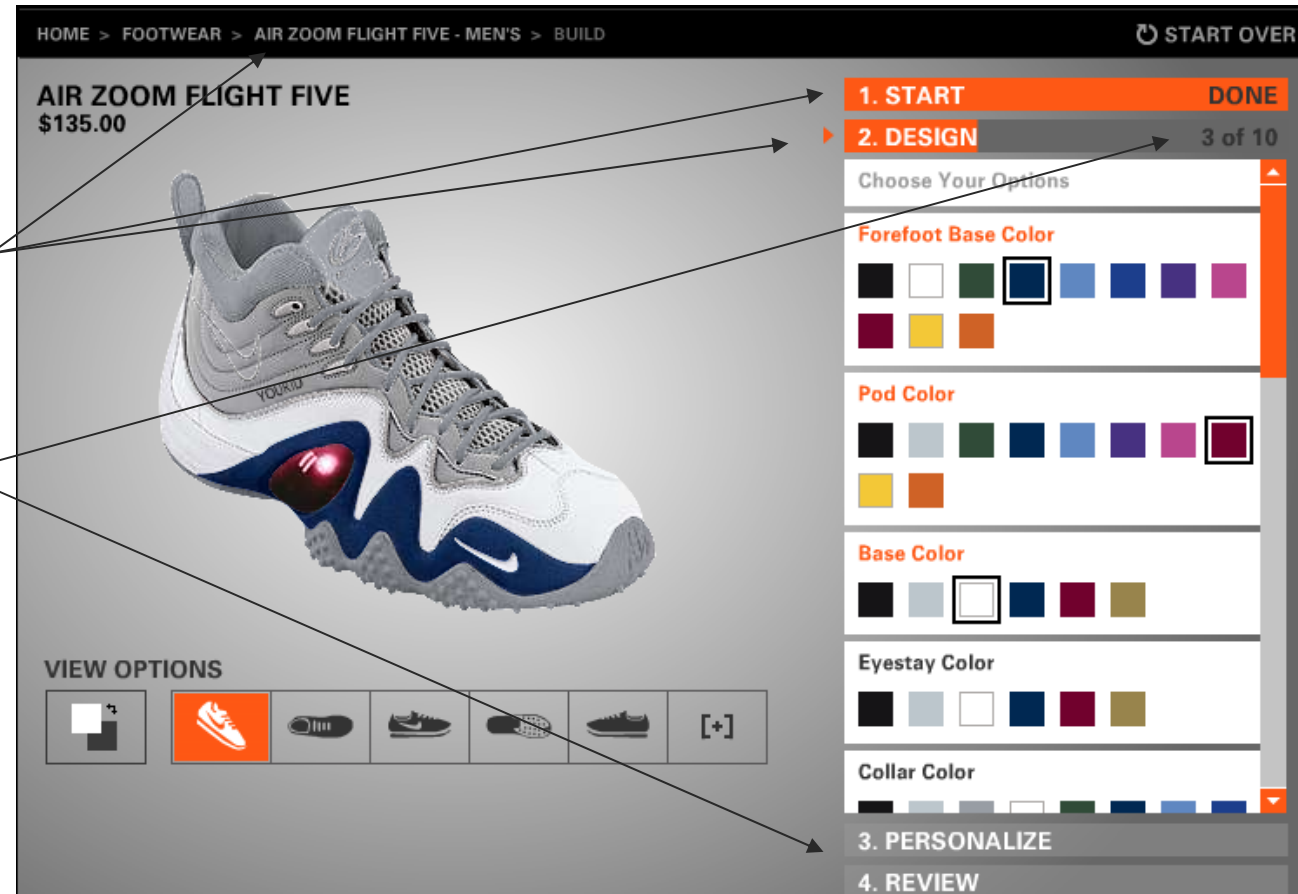
6. 依赖识别而非记忆



Example: NikeiD

Previous choices

What remains?





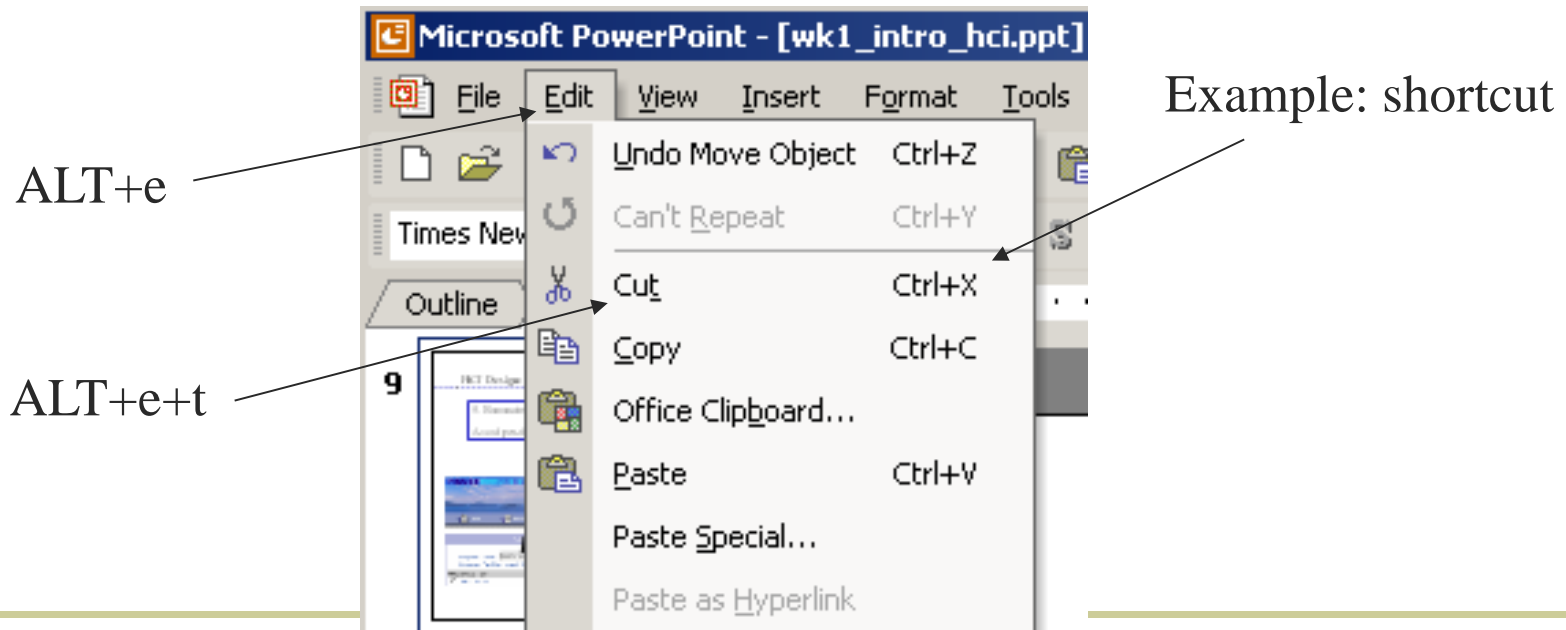
7. 使用的灵活性和高效性



For novice users, provide easy (though longer) interactions),

For advanced/frequent users, provide: short-cut, special keys, macros, ...

Example: Special keys





8. 帮助用户识别、诊断和恢复错误



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

Poor design examples:





9.帮助和文档



- Must provide help/manual/user-guide
- Language and format of User-guide should use simple, standard terminology

MS Help: (Good design)

- standardized format; provides search; book-metaphor; use of links





10. 审美感和最小化设计



- Do not put too much, irrelevant information in Dialog boxes
- Use standard and common controls (sliders, buttons etc.)
- Select fonts/sizes that are suitable for display to maximize readability

Fonts designed for web use

This is an example of Verdana

This is an example of Georgia

Others: Arial, Comic Sans MS



in Dialog boxes
controls (sliders, buttons etc.)

display to maximize readability

even in large/small font size):

web (Internet Explorer default)



小结



- 设计目标
 - 可用性目标
 - 用户体验目标
- 简易可用性工程
 - 可用性属性的度量
 - 四项关键技术
- 设计原则
 - 一般原则
 - 黄金规则
 - 启发式规则



作业



- 提交大作业题目和简要立项说明
 - 立项说明应包括：是什么、用户是谁、做什么、领域现状、可用性目标（不局限于课堂内容）
 - 本学期课程将**不要求**实现
 - 更强调原型设计、迭代过程、可用性评估
- 截止时间
 - 4月30日前