

2.3 交互技术、软件工程的介绍

交互设计所承载的媒介从最开始的电子产品到软件界面、网页，再到移动终端 APP、智能穿戴设备，一直都在不断演变，而这种演变来源于交互技术及软件工程的发展。

1.1.1 计算机界面的发展史

计算机界面的发展史大致可划分为四个时期：1960 年前，1960 年至 1984 年，1984 年后，2000 年前后。



2-71 穿孔卡计算机界面
2-71 Punched card computer

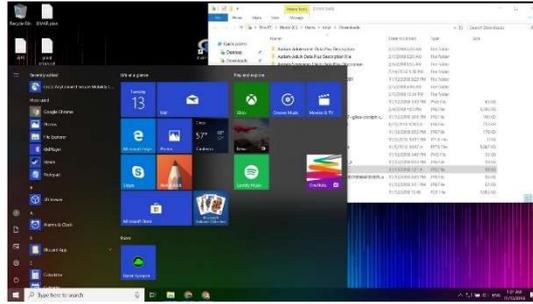
2-73 命令行界面：命令行界面，即命令语言解释器 (CLI)，也称为命令行用户界面、控制台用户界面或字符用户界面 (CUI)，是一种与计算机程序交互的方式，其中用户（或客户端）以连续文本行（命令行）的形式向程序发出命令。

```
Welcome to FreeBSD
DateMouse v1.9.1 alpha 1 (FreeBSD)
Installed at PS/2 port
C:\>ver
FreeCOM version 0.82 pl 3 XMS_Swap (Dec 10 2003 06:49:21)
C:\>dir
Volume in drive C is FREEBOS_C95
Volume Serial Number is 0E4F-19EB
Directory of C:\

FREEBOS      <DIR>    08-26-04  6:23p
FREEBOS\BAT  435     08-26-04  6:24p
FREEBOS\BIN  512     08-26-04  6:23p
FREEBOS\COM  93,963  08-26-04  6:24p
FREEBOS\SVS  881     08-26-04  6:24p
FREEBOS\BOOT 512     08-26-04  6:24p
FREEBOS\SYS  45,815  04-17-04  9:49p
6 file(s)    142,838 bytes
1 dir(s)    1,064,517,632 bytes free
C:\>
```

2-72 命令行界面
2-72 Command-line Interface

2-74 WIMP 界面模型：WIMP 分别代表：W - Windows（屏幕），I - Icons（图表），M - Menus（菜单），P - Pointing device（定位设备）。



2-73 WIMP 界面模型
2-73 WIMP Interface model

2-75 Post-WIMP 界面：具以下特点：无菜单，无窗口，无工具栏，以手势及语音识别为输入端。

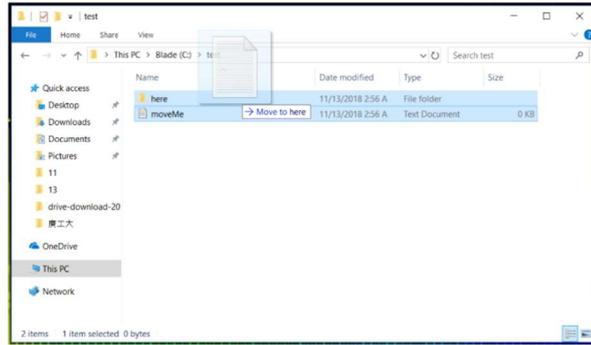


2-74 Post-WIMP 界面
2-74 Post-WIMP Interface

完美的计算机界面知道“你想要什么”以及“你做完了什么”。如何改进当今的界面设计方案介于这两个问题之间，使得人们工作时不会受界面的影响而分心！

1.1.2 WIMP 界面的优点和缺点

优点：影响计算机界面评价的最关键因素是这一界面的用户友好度。对于普通用户而言，这主要由学习界面操作并记忆的难易程度决定。而对于高级用户而言，他们更重视精通界面使用所需的时间成本，而非入门难度。虽然 WIMP 界面离真正完美的计算机界面尚有一定距离，但它已经有了一定进展。所有用户都能轻松使用这一界面，甚至是一些，通常来说，由于早期界面操作困难无法使用的用户如尚未学习读写的儿童和非专业用户。“指向并单击”是 WIMP 图形用户界面的标志，已成为现代计算机文化的一部分。用户界面设计已经成为一门专业且用户界面设计师们备受追捧。WIMP 图形用户界面使应用程序界面具有事实标准成为现实，与命令行界面相比，由于外观一致，这一界面使我们（相对）易于学习、易于使用，并且易于把一个应用程序获得的知识迁移到另一个应用程序。“再也没有人阅读说明书了”，毕竟，总的来说，他们再也不必阅读说明书了。



2-75 WIMP 界面 2-75 WIMP Interface

缺点：使用鼠标或键盘进行输入并不适用于所有用户，可能是因为他们不习惯使用鼠标和键盘，亦或是因为长期使用会引发重复性劳损，更不要说残障人士无法使用鼠标和键盘了。WIMP 界面在语音识别及触控输入尚有非常大的发展空间。



2-76 触控输入界面 2-76 touch input interface

比尔·巴克斯顿，加拿大计算机科学家及设计师，微软研究院首席研究员，作为人机交互领域的先驱之一而闻名。他指出，除了仅有单眼，几根单关节手指，缺乏其他感觉器官的生物，仅以键盘和鼠标为基准样式的 WIMP 图形用户界面都不完美。



2-77 多样化输入界面
2-77 Diversified input interface

1.1.3 Post-WIMP 界面

其具有几个特点：1、基于现实的交互设计：Post-WIMP 界面的框。2、朴素物理学：人们对物理世界有常识性认知。3、身体认知意识及控制能力：人们能够感知自己的身体，并拥有控制及协调身体的能力。4、环境认知意识及影响能力：人们能够感知周围环境，并具备影响环境，在环境中定位及移动的能力。5、社会认知意识及交流能力：人们通常能够感知到周边环境中的其他人并具备与他们交流的能力。



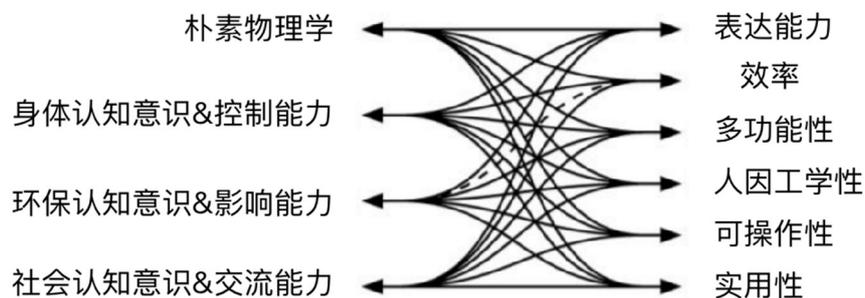
2-78 Post-WIMP 界面

2-78 Post-WIMP Interface

基于现实的交互应遵循以下设计原则：

- 表达能力：用户可以在应用界面内执行各种任务。
- 效率：用户可以快速执行任务。

- 多功能性：用户可以利用不同的应用程序执行各式各样的任务。
- 人因工学性：执行任务时，用户不会感到身体疲惫或对身体造成损伤。
- 可操作性：不论是什么样的用户都可以使用。
- 实用性：该系统具有后续开发和生产的潜力。



2-79 基于现实的交互设计权衡原则

2-79 Reality-Based Interaction design tradeoffs

图片出处：Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., & Zigelbaum, J. (2008, April). Reality-based interaction: a framework for post-WIMP interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 201-210).

1.1.4 软件工程介绍

软件的定义：计算机软件，也可以简称为软件，是数据的集合体或者是一种告知计算机如何工作的计算机指令。计算机软件与物理硬件相比，其系统是已构建完备的并且能直接执行指令。在计算机科学和软件工程中，计算机软件就是由计算机系统、程序和数据处理所整合的信息集合体。计算机软件包括计算机程序、相关信息库以及相关不可执行的数据，例如在线文档或数字媒体。计算机硬件和软件互为必需品并且两者都不能单独使用。软件的重要性体现在 1. 软件无处不在；2. 所有发达国家的经济发展都离不开软件开发；3. 软件开销通常主导系统开销；4. 软件的维护成本高于开发成本。



2-80 移动设备上的软件工程 2-80 software Even on mobile devices

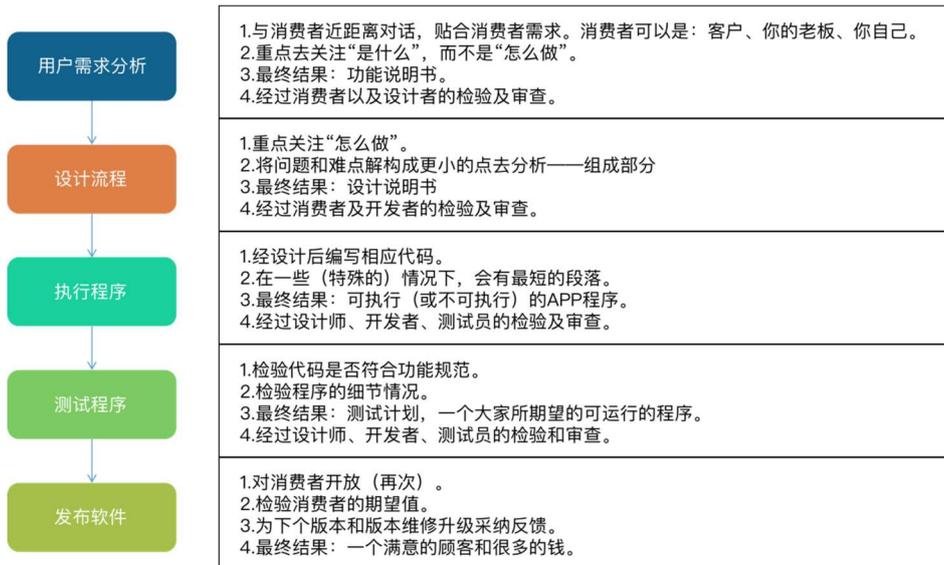
软件工程的定义。“工程”是指为了创建新产品或服务而去解释和排序必要任务或工作的作用形式。“软件工程”则是涉及各个方面的工程学科软件生产，捕获客户的业务需求并为系统指定蓝图，以便程序员实施。软件工程是应用系统的、规范的、可量化的方法来开发、操作和维护软件。软件工程师应该做到：对他们的工作采取系统性和组织性的方法；使用合适的工具和技术。这些措施都要基于：要解决的问题、发展的各种限制、可利用的资源。且都要有一定的预算支持，都要在所给期限前实行，都要在变化发生时及时执行。

软件开发过程中可能会遇到的一些问题：1. 所开发的软件不符合用户的使用需求；2. 部署后难以扩展；3. 贫瘠的文档资源；4. 开发质量较差；5. 开发过程中所耗费的资金和时间要比预期的还要多。

一个优秀的软件所具备的条件：1. 能提供用户所需要的功能和资源；2. 它是可持续发展的：不断更新迭代以满足用户不断变化、升级的使用需求；3. 它是可靠的：必须是让用户值得信赖和安全可靠的；4. 对于用户来说它是有求必应的；5. 它是高效的：开发者不应该浪费其系统资源；6. 它是可回收利用的：能以其他形式重复利用或者用于其他用途及目的。

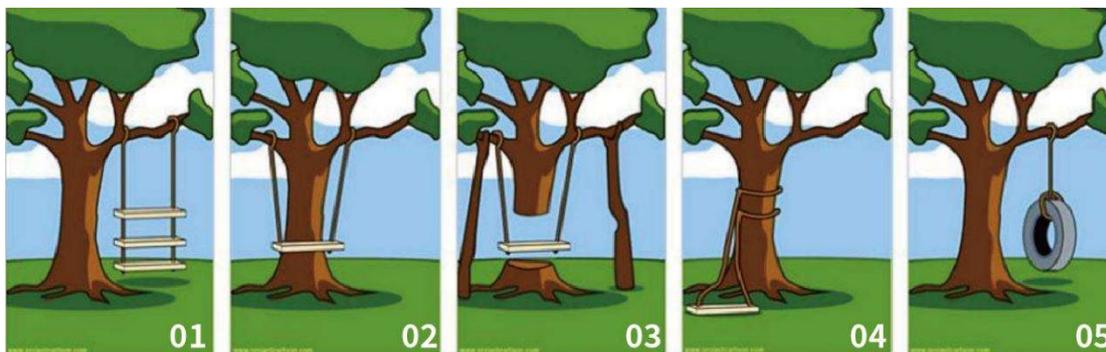
一个软件工程师所具备的技能：1. 软件工程师不只是一个开发者！2. “尽力而为是不够的：你必须知道需要做什么，然后再尽其所能做到最好”——W. Edwards Deming；3. 除了基础的技术技能，软件工程师还必须具备其他技能：沟通能力、及时反馈、工作积极性、分析能力、计划能力、风险应对能力。

SDLC：软件开发生命周期。它是一个创建和更改系统的过程，以及人们用来开发这些系统的模型和方法，在这个过程中团队内部主要由产品经理、开发者、设计师、测试员构成（但不仅限于这些成员）



2-81 软件开发生命周期 2-81 Software Development Lifecycle

软件开发风险：风险与设计师的关系如下所图示。P1：消费者如何去解释它。P2：产品主管如何去理解它。P3：分析者如何去设计它。P4：程序员如何去编写它。P5：什么是消费者真正需要的。

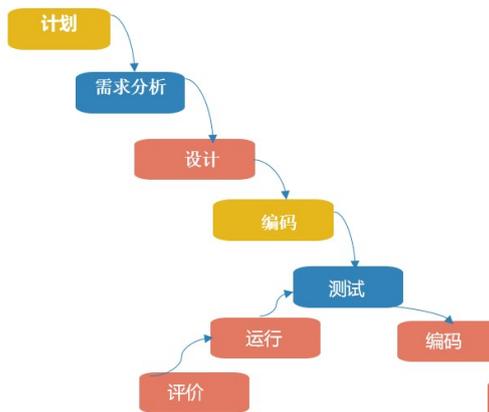


2-82 开发风险与设计师的关系 2-82 Disasters related to Designers

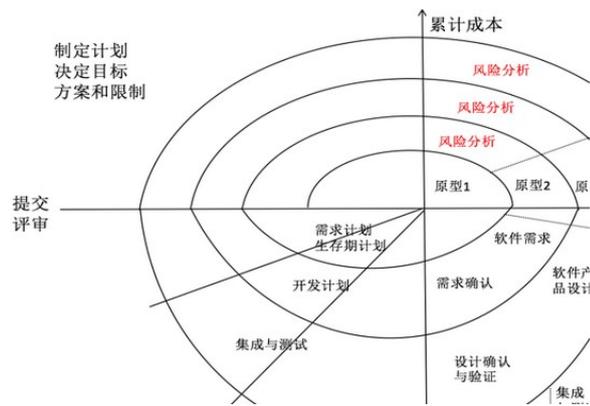
软件工程是为了保证生产的时效，防止错过最后期限；保证产品质量，防止系统崩溃；保证软件的完整性，有完整的文档并满足用户的需求。什么是软件运行过程模型？有许多不同的软件运行过程，但都涉及到：

- 规范性——定义系统应该做什么。
- 设计和实施——定义组织制度和实施制度。
- 验证——检查它是否符合客户的要求。
- 升级——改良系统以满足客户不断升级的需求。

软件开发生命周期模型：最常用、流行且重要的几种模型：瀑布模型、V模型、增量模型、RAD模型、敏捷模型、迭代模型、螺旋模型、原型模型。



2-83 瀑布模型
2-83 Waterfall Model



2-84 螺旋模型
2-84 Spiral Model