

| College of Computer Science, Chongqing University |

Software Engineering

A Practitioner's Approach Seventh Edition

12 设计概念

zmqmail@cqu.edu.cn 13708390417



对人造事物的构想与规划

Conception and Planning of the Artificial
(在上世纪，学界就已经对「设计」的公认解释)

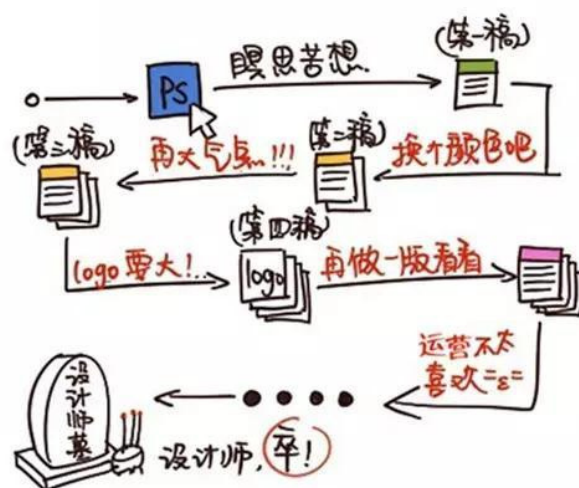
构想与规划(Conception and Planning)

- 设计过程描述为五个步骤(对这五个步骤的描述有很多版本，但其实说的都是一件事)：

- 发现并定义问题
- 通过调查与研究来深入理解问题
- 创造性地提出构想
- 精炼构想并付诸实践
- 进一步迭代



设计师现实



好的软件设计 (Good software design Design)

■ **Firmness** (坚固):

- A program should not have any bugs that inhibit its function.
程序应该不含任何妨碍其功能的缺陷

■ **Commodity** (适用):

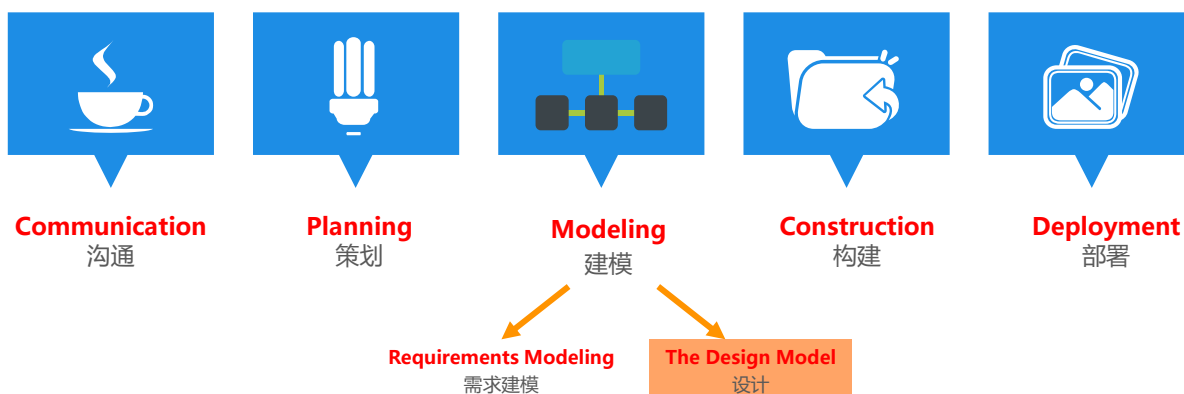
- A program should be suitable for the purposes for which it was intended.
程序符合开发目标

■ **Delight** (愉悦):

- The experience of using the program should be pleasurable one.
使用程序的体验是愉快的

PS: Mitch Kapor, the creator of Lotus 1-2-3, presented a "software design manifesto" in Dr. Dobbs Journal.

Process Model



一旦对软件需求进行分析和建模,软件设计就开始了。

12.1 软件工程中的设计

■ 软件设计在软件工程中处于核心技术

- Software design sits at the Technical Kernel of software engineering and is applied regardless of the software process model that is used.

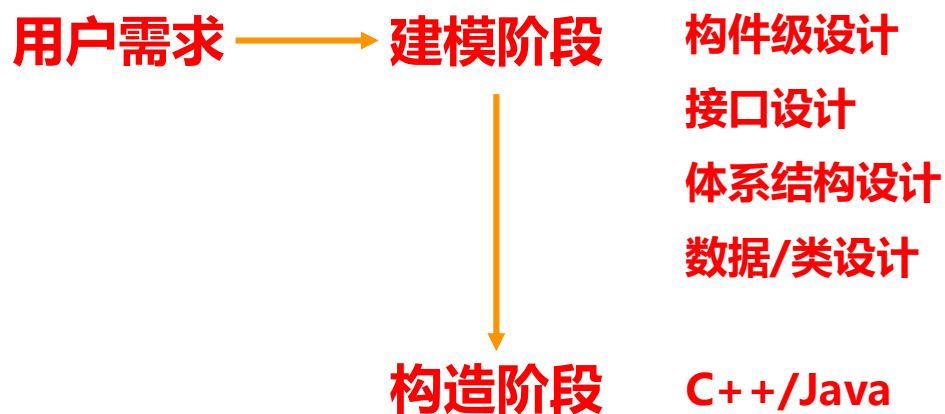
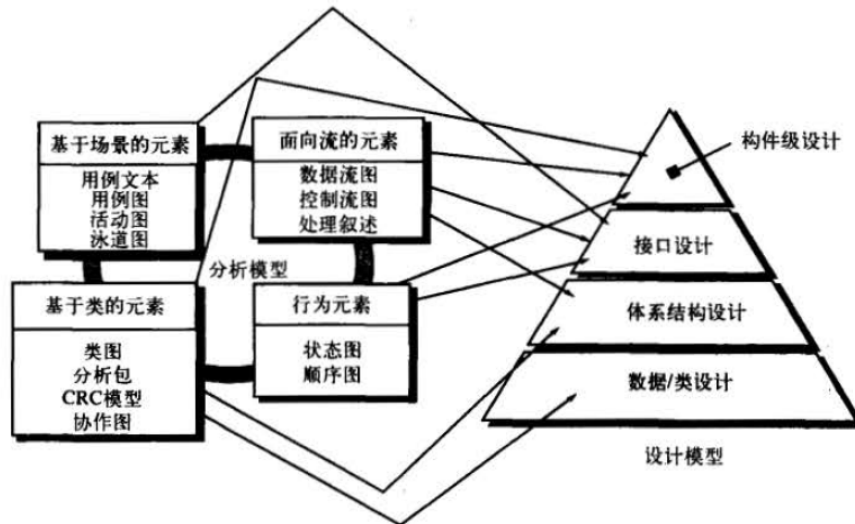
■ 建模活动的最后一个软件工程活动，其后就是软件的构造阶段(代码生成和测试)

- Beginning once software requirements have been analyzed and modeled, software design is the Last software engineering Action within the modeling activity and sets the stage for construction (code generation and testing).



建模活动的最后一个活动

需求模型->设计模型的转换



设计的概念 (Design Concepts)

- 设计模型提供软件体系结构、数据结构、接口和构件的细节

Design produces a data/class design, an architectural design, an interface design, and a component design

- 数据设计或类设计

- Data/Class Design

- 系统或产品的体系结构设计

- Architectural Design

- 接口设计

- Interface Design

- 构件级设计

- Component-level Design

技术评审

The Technical Review
Discussed in detail in Chapter 15



12.1 软件质量指导原则和属性

■ 良好设计演化的三个特征

■ 1 满足需求

- 设计必须实现所有包含在需求模型中的明确需求，而且必须满足利益相关者期望的所有隐含需求。

■ 2 设计可读、可理解

- 对于那些生成代码的人和那些进行测试以及随后维护软件的人而言，设计必须是可读的、可理解的指南。

■ 3 提供全貌

- 设计必须提供软件的全貌，从实现的角度说明数据域、功能域和行为域。

12.1 软件质量指导原则和属性

■ 质量指导原则(8条)

- 1. 设计应展示出这样一种结构: (a) 已经使用可识别的体系结构风格或模式创建 (b) 由能够展示出良好设计特征的构件构成(将在本章后面讨论) (c) 能够以演化的方式实现, 从而便于实现和测试。
- 2. 设计应该模块化, 也就是说, 应将软件逻辑地划分为元素或子系统。
- 3. 设计应该包含数据、体系结构、接口和构件的清晰表示。
- 4. 设计应导出数据结构, 这些数据结构适用于要实现的类, 并从可识别的数据模式提取。
- 5. 设计应导出显示独立功能特征的构件。
- 6. 设计应导出接口, 这些接口降低了构件之间以及与外部环境连接的复杂性。
- 7. 设计的导出应根据软件需求分析过程中在取的信息采用可重复的方法进行。
- 8. 应使用能够有效传达其意义的表示法来表达设计。

FURPS

(functionality, usability, reliability, performance, and supportability)

质量属性 (Quality Attributes)

- **FURPS**(functionality, usability, reliability, performance, and supportability)
- **Functionality(功能性)**
 - 通过评估程序的特征集和能力、所提交功能的通用性以及整个系统的安全性来评估
- **Usability(易用性)**
 - 通过考虑人员因素、整体美感、一致性和文档来评估
- **Reliability(可靠性)**
 - 通过测量故障的频率和严重性、输出结果的精确性、平均故障时间、故障恢复能力和程序的可预见性来评估
- **Performance(性能)**
 - 通过考虑处理速度、响应时间、资源消耗、吞吐量和效率来度量
- **Supportability(可支持性)**
 - 综合可扩展性、适应性和耐用性——可维护性!

12.3 设计概念(设计的方法)(Fundamental Concepts)

- **抽象 Abstraction**—data, procedure, control
- **体系结构 Architecture**—the overall structure of the software
- **模式 Patterns**—“conveys the essence” of a proven design solution
- **关注点分离 Separation of concerns**—any complex problem can be more easily handled if it is subdivided into pieces
- **模块化 Modularity**—compartmentalization of data and function
- **信息隐藏 Hiding**—controlled interfaces

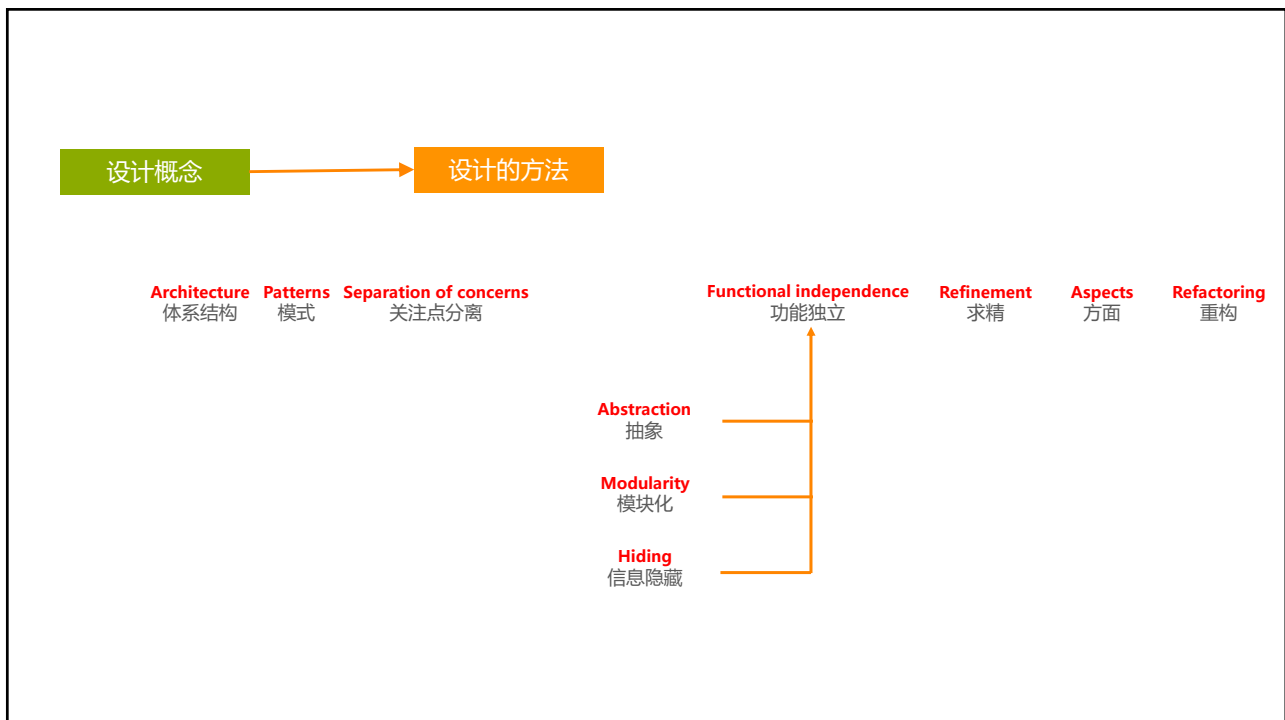
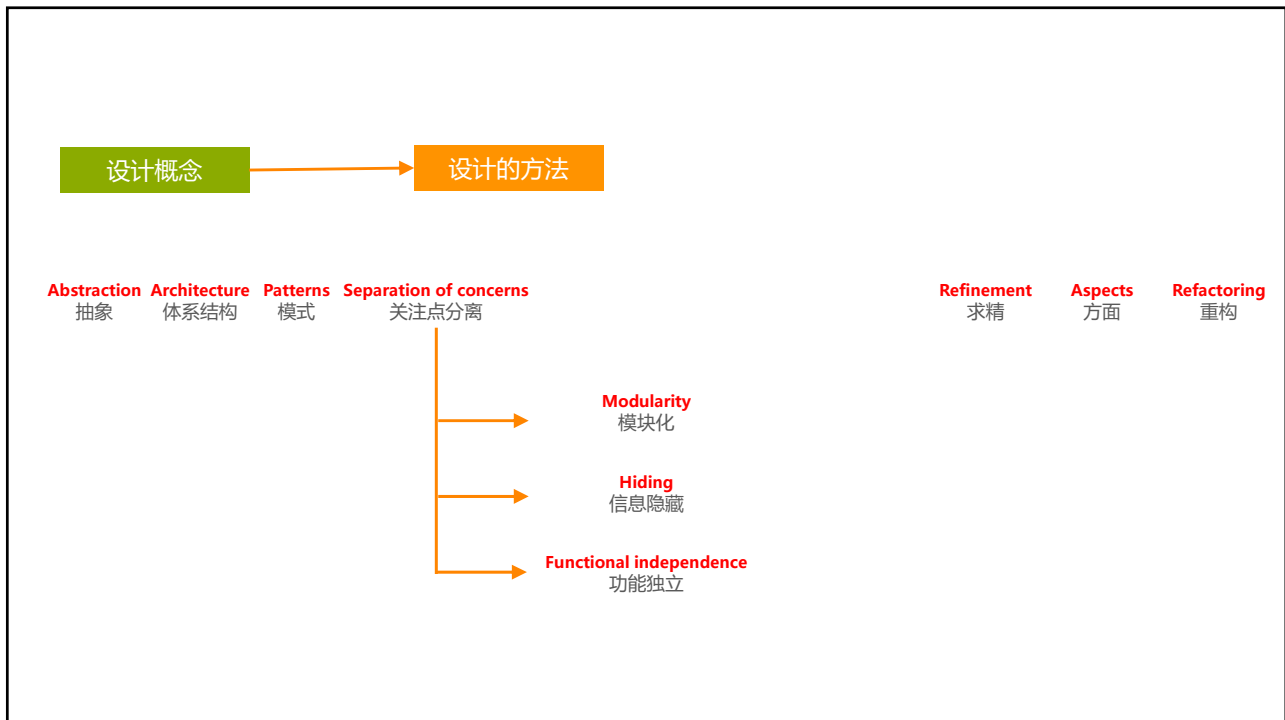
12.3 设计概念(设计的方法)(Fundamental Concepts)

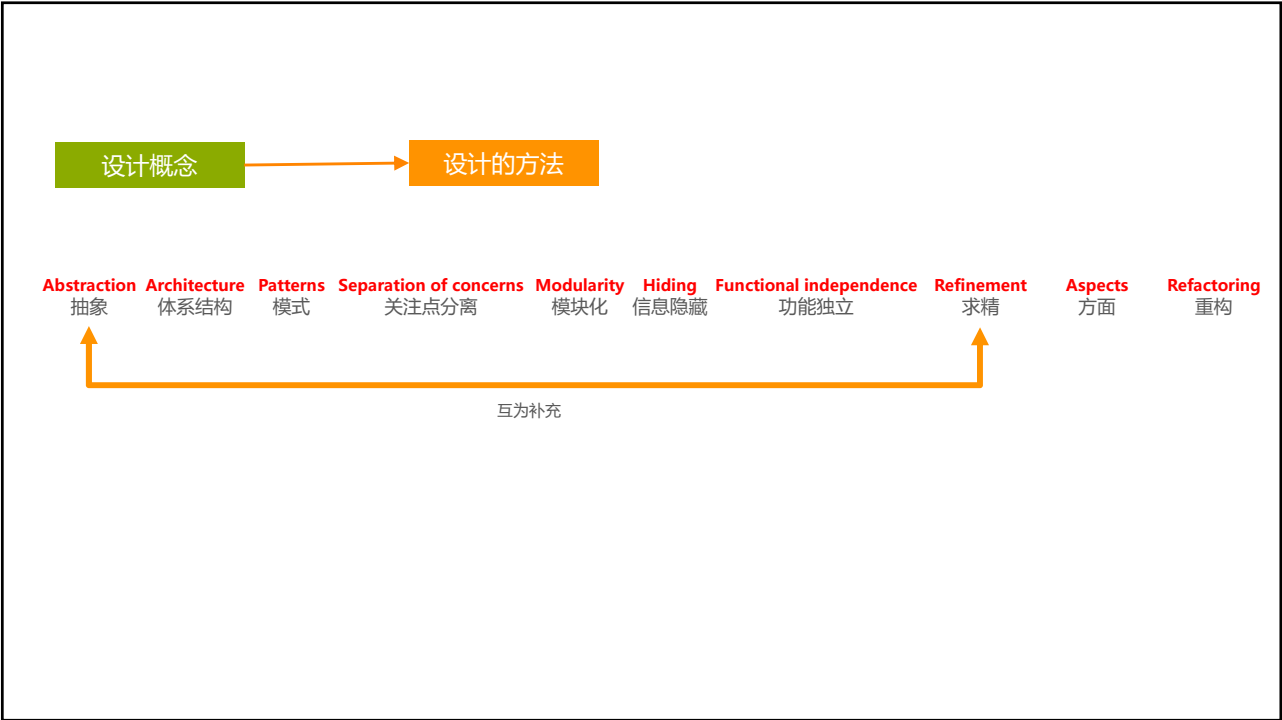
- **功能独立 Functional independence**
 - single-minded function and low coupling
- **求精 Refinement**—elaboration of detail for all abstractions
- **方面 Aspects**
 - a mechanism for understanding how global requirements affect design
- **重构 Refactoring**
 - a reorganization technique that simplifies the design
- **OO design** concepts—Appendix II
- **Design Classes**—provide design detail that will enable analysis classes to be implemented

设计概念

设计的方法

Abstraction 抽象 Architecture 体系结构 Patterns 模式 Separation of concerns 关注点分离 Modularity 模块化 Hiding 信息隐藏 Functional independence 功能独立 Refinement 求精 Aspects 方面 Refactoring 重构





阿里高级技术专家方法论：如何写复杂业务代码？

从码农到工匠 阿里技术 2019-08-14

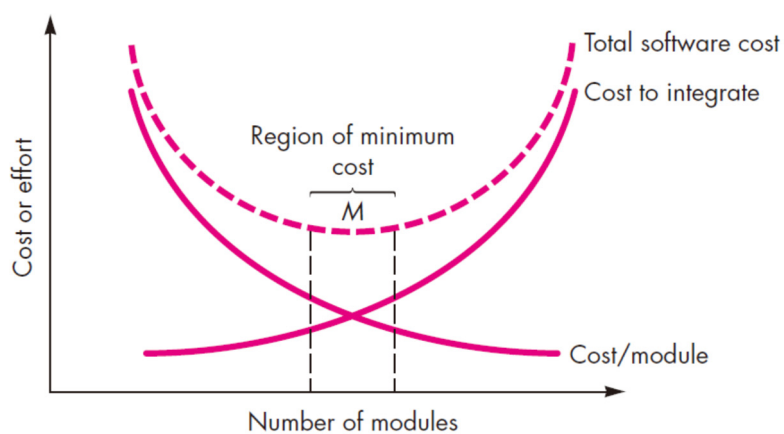


阿里妹导读：张建飞是阿里巴巴高级技术专家，一直在致力于应用架构和代码复杂度的治理。最近，他在看零售通商品域的代码。面对零售通如此复杂的业务场景，如何在架构和代码层面进行应对，是一个新课题。结合实际的业务场景，Frank 沉淀了一套“如何写复杂业务代码”的方法论，在此分享给大家，相信同样的方法论可以复制到大部分复杂业务场景。

代码还是要脚踏实地

写够行数!

■ 模块化和软件成本 Modularity: Trade-offs



Modularity and software cost

信息隐藏(Information Hiding)

■ 好的模块设计的一个基本特征

- 它指的是限制软件系统的指定部分的信息。
- 信息隐藏的关键就是阻止模块的动作，使其不会对其他模块产生不必要的依赖或影响。
 - 否则，就有可能导致模块产生错误，这些错误可能是在其他模块的开发中带来的，亦或是在软件维护期间不正确的维护带来的。

■ 信息隐藏是抽象主题和抽象工具使用的中心。

- 抽象工具的概念是“黑盒”的概念
- 用户可以忽略它的内部特性，允许用户集中考虑手头更大的应用。
 - 就像安全罩可以用来防护电子设备复杂的、潜在的风险一样。
 - 保护他们的用户远离内部危险，同样也保护内部，以防来自其他用户的侵扰。

功能独立性

■ 功能独立性是抽象、模块化和信息隐蔽的直接产物。如果一个模块能够独立于其他模块被编程、测试和修改，则该模块具有功能独立性。

- 1978年Myers提出用**两个准则**来度量功能独立性，即模块间的耦合和模块的内聚。

■ 内聚性:

- 内聚是一个模块内部各个元素彼此结合的紧密程度的度量。

■ 耦合性:

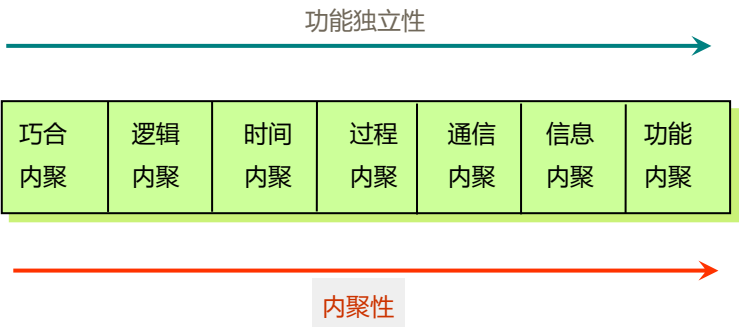
- 耦合是模块间互相连接的紧密程度的度量，它取决于各个模块之间接口的复杂度、调用方式以及哪些信息通过接口。

高内聚，低耦合。

要求模块结构达到高内聚，低耦合。

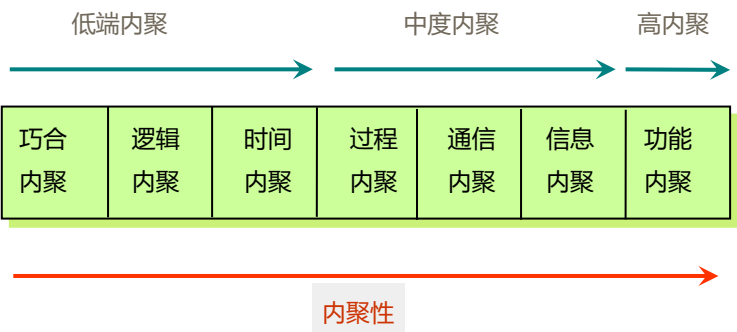
模块内聚

- 要求模块结构达到**高内聚，低耦合**。
- 模块内聚性越强，功能独立性越好，对于形成的模块结构有比较好的作用。



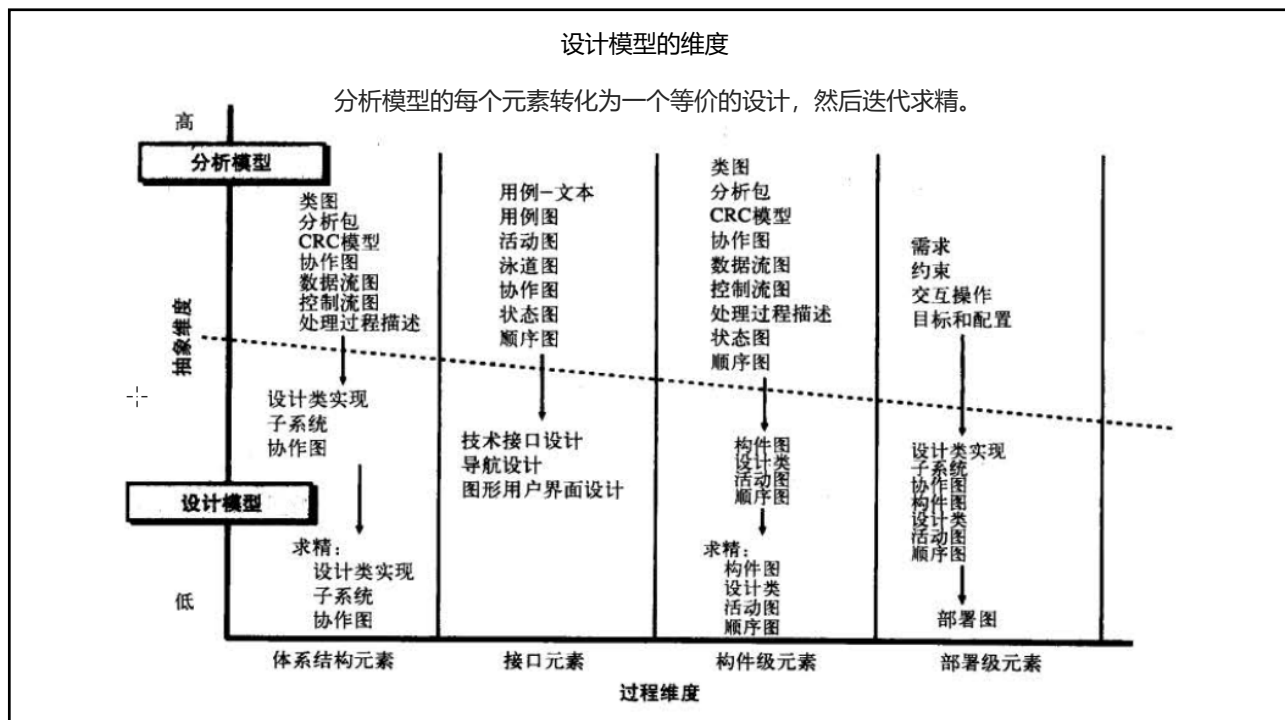
模块内聚

- 要求模块结构达到**高内聚，低耦合**。
- 模块内聚性越强，功能独立性越好，对于形成的模块结构有比较好的作用。



消除重复，分离关注点，统一抽象层次

指导原则：消除重复，分离关注点，统一抽象层次



设计模型的元素 Design Model Elements

数据设计元素 Data elements

- Data model --> data structures
- Data model --> database architecture

体系结构设计元素 Architectural elements

- Application domain
- Analysis classes, their relationships, collaborations and behaviors are transformed into design realizations
- Patterns and "styles" (Chapters 9 and 12)

Ch13 体系结构设计

接口设计元素 Interface elements

- The user interface (UI)
- External interfaces to other systems, devices, networks or other sources of information
- Internal interfaces between various design components.

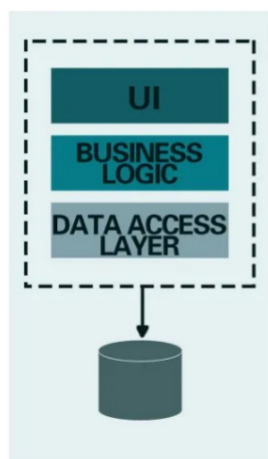
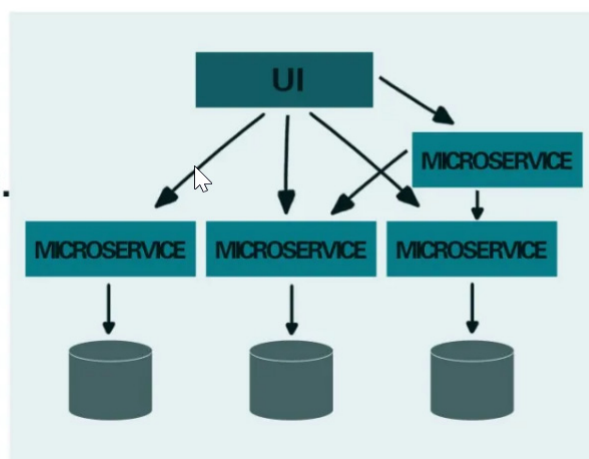
Ch14 构件级设计

Ch15 用户界面设计

构件级设计元素 Component elements

Ch14 构件级设计

部署级设计元素 Deployment elements

MONOLITHIC**VS.****MICROSERVICES****原型设计的基本概念和工具**

分享



原型设计需要表达清楚这么几个地方：

- **1、界面元素**

- 什么是界面元素，比如文字，下拉框，按钮，图标、图片等等这些都属于界面元素。

- **2、数据逻辑**

- 比如一个直播列表界面，上面有三个tab，分别是关注直播、热门直播、最新直播，那么最新直播是基于怎样的数据逻辑获取的，你就需要在你的原型设计上进行说明了。

- **3、操作逻辑**

- 一个原型界面上可以进行操作元素的有哪些，哪个可以点击，哪个可以选择，操作后出现怎样的反馈，比如弹出浮层？进入新页面？或是跳出新页面？还是给一个怎样的提示？这些也是需要在原型设计里面说清楚的。

原型图的 “道” “法” “器”

原型图的格式，属于 “道” ， “法” ， “器” ， 这3个层次中的 “器” 。



产品的“粘性”——文艺线路

重视情感化设计

重视情感化设计

重视情感化设计，通过功能与细节设计的巧思，为用户带来惊喜、温暖与愉悦。

重视情感化设计

- 在专注于认知层面，做出好用实用满足用户痛点的产品基础上，还应关照用户的情感需求与情绪感知。
- 重视情感化设计，通过功能与细节设计的巧思，为用户带来惊喜、温暖与愉悦。
- 能引发用户情感共鸣的产品，才是完整且能够俘获人心的好产品。



原型设计工具

工欲善其事必先利其器！

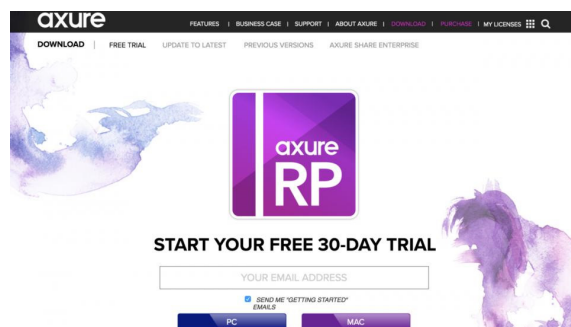


Axure RP

推荐指数 ◆◆◆◆◆

Axure RP是一个专业的快速原型设计工具。Axure(发音:Ack-sure),代表美国Axure公司;RP则是Rapid Prototyping(快速原型)的缩写。

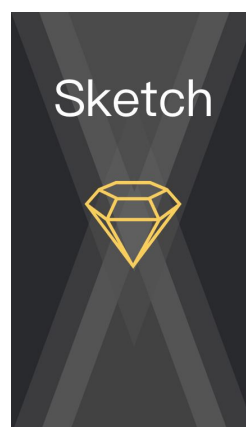
Axure是产品经理的必备工具,其他的不用说了,去看看招聘网站上关于PM的招聘信息,基本所有职位描述里,都包含了这么一句“熟练使用Axure”,所以你懂的。



Sketch

推荐指数 ◆◆◆◆◆

一款轻量, 易用的矢量设计工具。



墨刀

推荐指数 ◆◆◆◆

墨刀也是一款原型设计软件，目前分为网页版和客户端，网页版可以直接使用，客户端其实也是网页版的功能，一般来说会下载客户端，使用过程中需要联网。

缺点也很明显：1、依赖网络，网络不稳定或者服务器不稳定的时候则没法使用；2、很多交互功能不够强大；



Visio

推荐指数 ◆◆◆◆

很多老牌的产品策划人员、产品经理都使用过它来设计产品原型，算是线框图专业户，比较灵活，Office使用习惯接受程度较高。



PowerPoint

推荐指数 ◆◆◆◆◆

PowerPoint2013有着全新的外观界面、更加简洁，而且平板电脑和手机等移动PC都可以使用。



Photoshop

Adobe



■ ■ ■ ■ ■ ■ ■ ■

| College of Computer Science, Chongqing University |

Software Engineering

A Practitioner's Approach Seventh Edition

12 设计概念

zmqmail@cqu.edu.cn 13708390417

