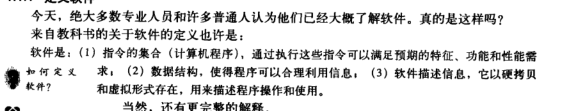
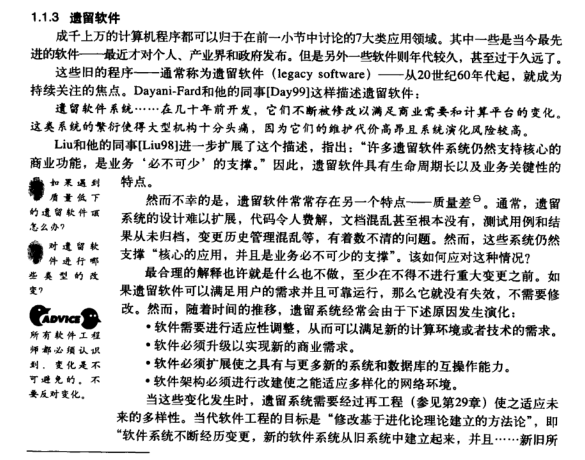
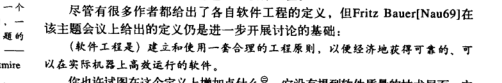
1什么是软件



2遗留系统

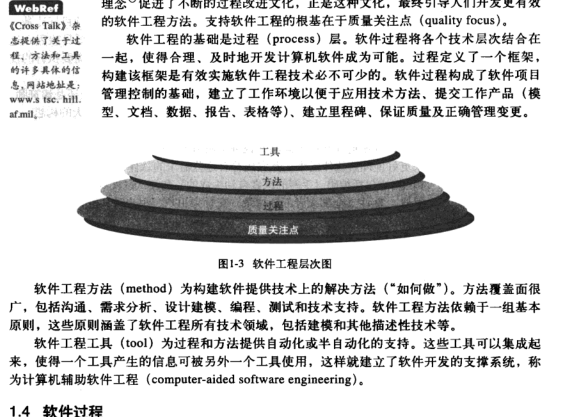


3软件工程作用

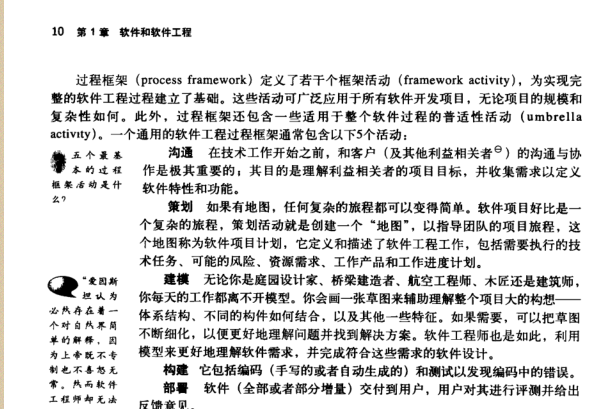




4软件开发过程三要素

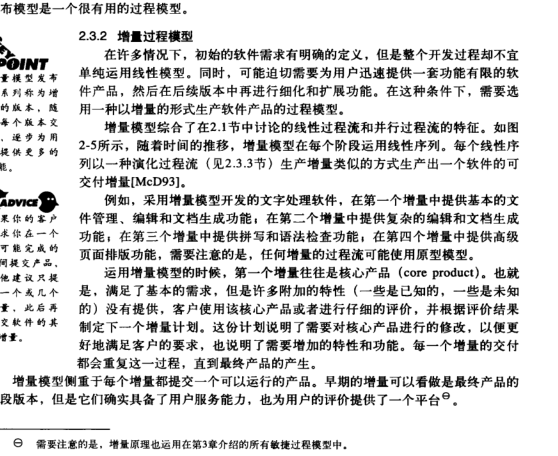
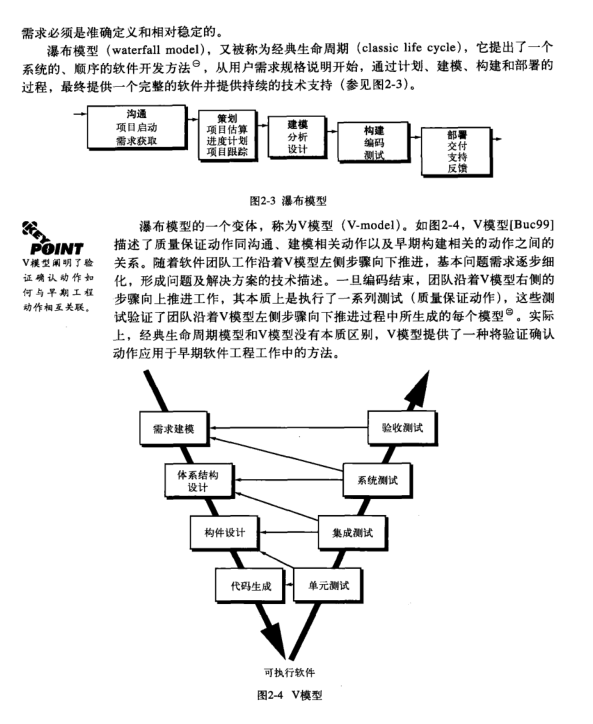


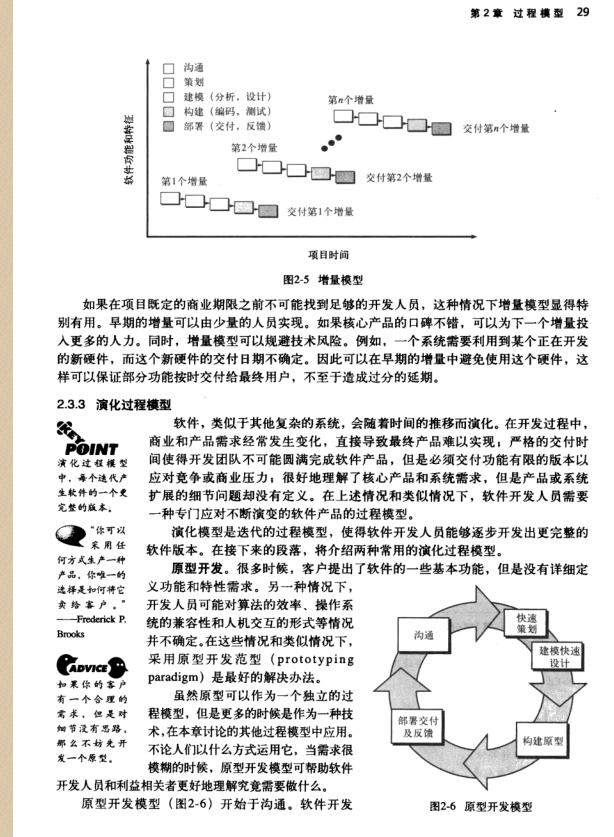
5五中框架活动

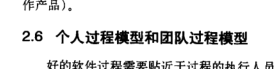
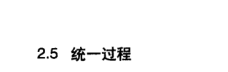
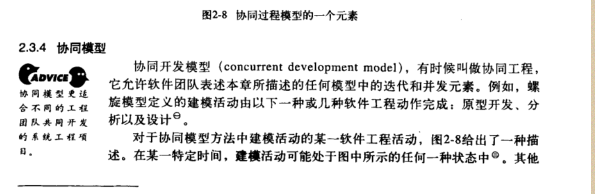
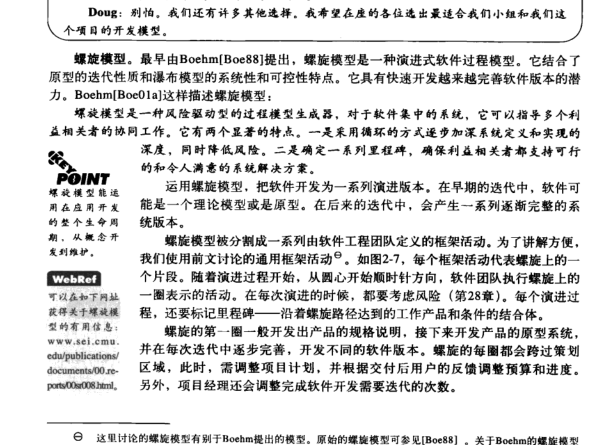


6惯例模型的种类









7软件开发各阶段的主要任务

### 1计划

对所要解决的问题进行总体定义，包括了解用户的要求及现实环境，从技术、经济和社会因素等3个方面研究并论证本[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)项目的可行性，编写可行性研究报告，探讨解决问题的方案，并对可供使用的资源（如[计算机硬件](http://baike.baidu.com/view/1024731.htm" \t "http://baike.baidu.com/_blank)、系统软件、人力等）成本，可取得的效益和[开发](http://baike.baidu.com/view/522596.htm" \t "http://baike.baidu.com/_blank)进度作出估计，制订完成开发任务的实施计划。

### 2分析

[软件需求分析](http://baike.baidu.com/view/671802.htm" \t "http://baike.baidu.com/_blank)就是对开发什么样的软件的一个系统的分析与设想。它是一个对用户的需求进行去粗取精、去伪存真、正确理解，然后把它用[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)工程开发语言（形式功能规约，即需求规格说明书）表达出来的过程。本阶段的基本任务是和用户一起确定要解决的问题，建立[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)的[逻辑模型](http://baike.baidu.com/view/1650561.htm" \t "http://baike.baidu.com/_blank)，编写需求规格说明书文档并最终得到用户的认可。[需求分析](http://baike.baidu.com/view/111493.htm" \t "http://baike.baidu.com/_blank)的主要方法有[结构化分析方法](http://baike.baidu.com/view/1441153.htm" \t "http://baike.baidu.com/_blank)、[数据流程图](http://baike.baidu.com/view/170104.htm" \t "http://baike.baidu.com/_blank)和[数据字典](http://baike.baidu.com/view/70173.htm" \t "http://baike.baidu.com/_blank)等方法。本阶段的工作是根据需求说明书的要求，设计建立相应的[软件系统](http://baike.baidu.com/view/8343.htm" \t "http://baike.baidu.com/_blank)的[体系结构](http://baike.baidu.com/view/1188494.htm" \t "http://baike.baidu.com/_blank)，并将整个系统分解成若干个子系统或模块，定义子系统或模块间的接口关系，对各子系统进行具体设计定义，编写软件[概要设计](http://baike.baidu.com/view/551728.htm" \t "http://baike.baidu.com/_blank)和[详细设计说明书](http://baike.baidu.com/view/624724.htm" \t "http://baike.baidu.com/_blank)，数据库或[数据结构](http://baike.baidu.com/view/9900.htm" \t "http://baike.baidu.com/_blank)设计说明书，组装[测试计划](http://baike.baidu.com/view/651486.htm" \t "http://baike.baidu.com/_blank)。在任何[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)或系统开发的初始阶段必须先完全掌握用户需求，以期能将紧随的系统开发过程中哪些功能应该落实、采取何种规格以及设定哪些限制优先加以定位。[系统工程师](http://baike.baidu.com/view/2057361.htm" \t "http://baike.baidu.com/_blank)最终将据此完成[设计方案](http://baike.baidu.com/view/3642811.htm" \t "http://baike.baidu.com/_blank)，在此基础上对随后的程序开发、系统功能和性能的描述及限制作出定义。

### 3设计

[软件设计](http://baike.baidu.com/view/575391.htm" \t "http://baike.baidu.com/_blank)可以分为[概要设计](http://baike.baidu.com/view/551728.htm" \t "http://baike.baidu.com/_blank)和详细设计两个阶段。实际上[软件设计](http://baike.baidu.com/view/575391.htm" \t "http://baike.baidu.com/_blank)的主要任务就是将软件分解成模块是指能实现某个功能的数据和程序说明、可执行程序的[程序单元](http://baike.baidu.com/view/2610106.htm" \t "http://baike.baidu.com/_blank)。可以是一个[函数](http://baike.baidu.com/view/15061.htm" \t "http://baike.baidu.com/_blank)、过程、[子程序](http://baike.baidu.com/view/933538.htm" \t "http://baike.baidu.com/_blank)、一段带有程序说明的独立的程序和数据，也可以是可组合、可分解和可更换的功能单元。模块，然后进行[模块设计](http://baike.baidu.com/view/2414000.htm" \t "http://baike.baidu.com/_blank)。[概要设计](http://baike.baidu.com/view/551728.htm" \t "http://baike.baidu.com/_blank)就是结构设计，其主要目标就是给出[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)的模块结构，用[软件结构图](http://baike.baidu.com/view/3224863.htm" \t "http://baike.baidu.com/_blank)表示。详细设计的首要任务就是设计模块的程序流程、算法和[数据结构](http://baike.baidu.com/view/9900.htm" \t "http://baike.baidu.com/_blank)，次要任务就是设计[数据库](http://baike.baidu.com/view/1088.htm" \t "http://baike.baidu.com/_blank)，常用方法还是[结构化程序设计方法](http://baike.baidu.com/view/260077.htm" \t "http://baike.baidu.com/_blank)。

### 4编码

[软件编码](http://baike.baidu.com/view/178766.htm" \t "http://baike.baidu.com/_blank)是指把[软件设计](http://baike.baidu.com/view/575391.htm" \t "http://baike.baidu.com/_blank)转换成[计算机](http://baike.baidu.com/view/3314.htm" \t "http://baike.baidu.com/_blank)可以接受的程序，即写成以某一程序设计语言表示的“源程序清单”。充分了解[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)开发语言、工具的特性和编程风格，有助于开发工具的选择以及保证软件产品的开发质量。

当前[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)开发中除在专用场合，已经很少使用二十世纪80年代的[高级语言](http://baike.baidu.com/view/14900.htm" \t "http://baike.baidu.com/_blank)了，取而代之的是[面向对象](http://baike.baidu.com/view/125370.htm" \t "http://baike.baidu.com/_blank)的开发语言。而且[面向对象](http://baike.baidu.com/view/125370.htm" \t "http://baike.baidu.com/_blank)的开发语言和开发环境大都合为一体，大大提高了开发的速度。

### 5测试

[软件测试](http://baike.baidu.com/view/16563.htm" \t "http://baike.baidu.com/_blank)的目的是以较小的代价发现尽可能多的错误。要实现这个目标的关键在于[设计](http://baike.baidu.com/view/14417.htm" \t "http://baike.baidu.com/_blank)一套出色的测试用例（测试数据与功能和预期的输出结果组成了[测试用例](http://baike.baidu.com/view/106882.htm" \t "http://baike.baidu.com/_blank)）。如何才能设计出一套出色的测试用例，关键在于理解测试方法。不同的测试方法有不同的[测试用例设计](http://baike.baidu.com/view/1270753.htm" \t "http://baike.baidu.com/_blank)方法。两种常用的测试方法是[白盒法](http://baike.baidu.com/view/302745.htm" \t "http://baike.baidu.com/_blank)测试对象是[源程序](http://baike.baidu.com/view/546605.htm" \t "http://baike.baidu.com/_blank)，依据的是程序内部的的[逻辑结构](http://baike.baidu.com/view/540423.htm" \t "http://baike.baidu.com/_blank)来发现[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)的编程错误、结构错误和数据错误。结构错误包括逻辑、[数据流](http://baike.baidu.com/view/166248.htm" \t "http://baike.baidu.com/_blank)、初始化等错误。用例设计的关键是以较少的用例覆盖尽可能多的[内部程序](http://baike.baidu.com/view/491248.htm" \t "http://baike.baidu.com/_blank)逻辑结果。白盒法和黑盒法依据的是软件的功能或软件行为描述，发现软件的接口、功能和结构错误。其中接口错误包括内部/外部接口、资源管理、[集成化](http://baike.baidu.com/view/2236379.htm" \t "http://baike.baidu.com/_blank)以及系统错误。黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。

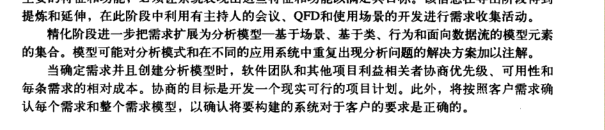
### 6维护

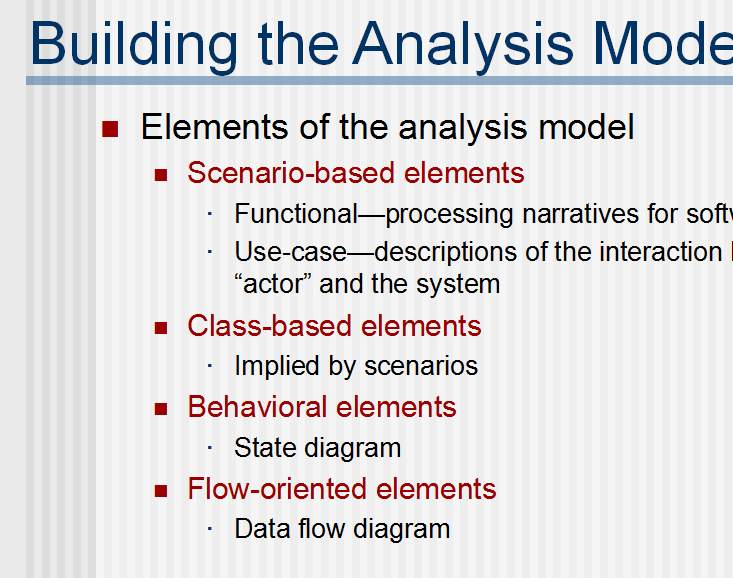
维护是指在已完成对软件的研制（分析、设计、编码和[测试](http://baike.baidu.com/view/1619.htm" \t "http://baike.baidu.com/_blank)）工作并交付使用以后，对软件产品所进行的一些[软件工程](http://baike.baidu.com/view/1659.htm" \t "http://baike.baidu.com/_blank)的活动。即根据软件运行的情况，对软件进行适当修改，以适应新的要求，以及纠正运行中发现的错误。编写[软件](http://baike.baidu.com/view/37.htm" \t "http://baike.baidu.com/_blank)问题报告、软件修改报告。

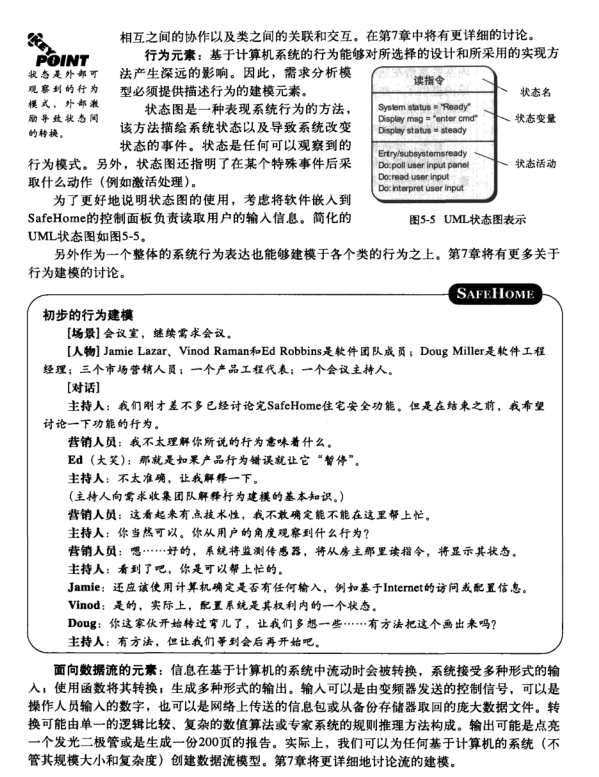
一个中等规模的软件，如果研制阶段需要一年至二年的时间，在它投入使用以后，其运行或工作时间可能持续五年至十年。那么它的维护阶段也是运行的这五年至十年期间。在这段时间，人们几乎需要着手解决研制阶段所遇到的各种问题，同时还要解决某些维护工作本身特有的问题。做好软件维护工作，不仅能排除障碍，使软件能正常工作，而且还可以使它扩展功能，提高性能，为用户带来明显的经济效益。然而遗憾的是，对[软件维护](http://baike.baidu.com/view/553458.htm" \t "http://baike.baidu.com/_blank)工作的重视往往远不如对软件研制工作的重视。而事实上，和软件研制工作相比，软件维护的工作量和成本都要大得多。

在实际开发过程中，软件开发并不是从第一步进行到最后一步，而是在任何阶段，在进入下一阶段前一般都有一步或几步的回溯。在[测试过程](http://baike.baidu.com/view/651488.htm" \t "http://baike.baidu.com/_blank)中的问题可能要求修改设计，用户可能会提出一些需要来修改需求说明书等。

8分析模型包括的元素







9对象的概念特征



继承多态，封装唯一

10结构化的方法，面向对象的方法

## **1.结构化方法遵循的基本原则**

结构化方法的基本思想就是将待解决的问题看作一个系统从而用系统科学的思想方法来分析和解决问题结构化方法遵循以下基本原则

###### **（1）抽象原则**

抽象原则是一切系统科学方法都必须遵循的基本原则它注重把握系统的本质内容而忽略与系统当前目标无关的内容它是一种基本的认知过程和思维方式

###### **（2）分解原则**

分解原则是结构化方法中最基本的原则它是一种先总体后局部的思想原则在构造信息系统模型时它采用自顶向下分层解决的方法

###### **（3）模块化原则**

模块化是结构化方法最基本的分解原则的具体应用它主要出现在结构化设计阶段中其目标是将系统分解成具有特定功能的若干模块从而完成系统指定的各项功能

## **2.面向对象模型遵循的基本原则**

面向对象模型遵循的基本原则有：抽象、封装、模块化以及层次原则等

###### **（1）抽象**

抽象是处理现实世界复杂性的最基本方式在OO方法中它强调一个对象和其他对象相区别的本质特性对于一个给定的域确定合理的抽象集是面向对象建模的关键问题之一

###### **（2）封装**

封装是对抽象元素的划分过程抽象由结构和行为组成封装用来分离抽象的原始接口和它的执行

封装也称为信息隐藏Information Hiding它将一个对象的外部特征和内部的执行细节分割开来并将后者对其他对象隐藏起来

###### **（3）模块化**

模块化是已经被分为一系列聚集的和耦合的模块的系统特性对于一个给定的问题确定正确的模块集几乎与确定正确的抽象集一样困难通常每个模块应该足够简单以便能够被完整地理解

###### **（4）层次**

抽象集通常形成一个层次，层次是对抽象的归类和排序。在复杂的现实世界中有两种非常重要的层次一个是类型层次另一个是结构性层次 。确定抽象的层次是基于对象的继承，它有助于在对象的继承中发现抽象间的关系，搞清问题的所在理解问题的本质

## **3.结构化方法的核心问题**

模型问题是结构化方法的核心问题建立模型简称建模是为了更好地理解我们要模拟的现实世界建模通常是从系统的需求分析开始在结构化方法中就是使用SA方法构建系统的环境模型然后使用SD方法确定系统的行为和功能模型最后使用SP方法进行系统的设计并确定用户的现实模型

## **4..面向对象方法的核心问题**

面向对象方法与结构化方法一样其核心问题也是模型问题面向对象模型主要由OOA模型OOD模型组成其中OOA主要属于学科抽象形态方面的内容OOD主要属于学科设计形态方面的内容

## **5．结构化系统分析方法的特点**

（1）面向用户，用户自始至终参与系统的分析工作；

（2）强调调查工作的重要性；

（3）对管理业务中的各种数据进行分解；

（4）采用了层次分解的系统思想；

（5）用图形工具来分析和构建新方案。

¨结构化分析的图表工具主要由数据流程图、数据字典和数据处理说明组成。

## **6.面向对象系统分析方法的特点**

**（1）** **强调系统开发的整体性和全局性**

**（2）** **侧重于数据转换的过程而不是数据本身**

**（3）** **系统的开发周期长**

**（4）** **封装性**

对象的概念突破了传统数据与操作分离的模式。对象作为独立存在的实体，将自由数据和操作封闭在一起，使自身的状态、行为局部化。

**（5）** **继承性**

　　继承是面向对象特有的，亦是最有力的机制。通过类继承可以弥补由封装对象而带来的诸如数据或操作冗余的问题，通过继承支持重用，实现软件资源共享、演化以及增强扩充。

**（6）** **多态性**

　　同一消息发送至不同类或对象可引起不同的操作，使软件开发设计更便利，编码更灵活。

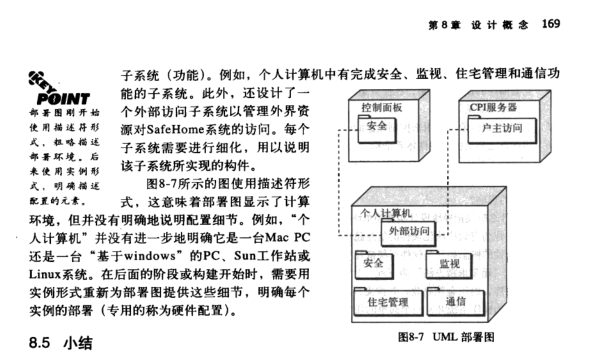
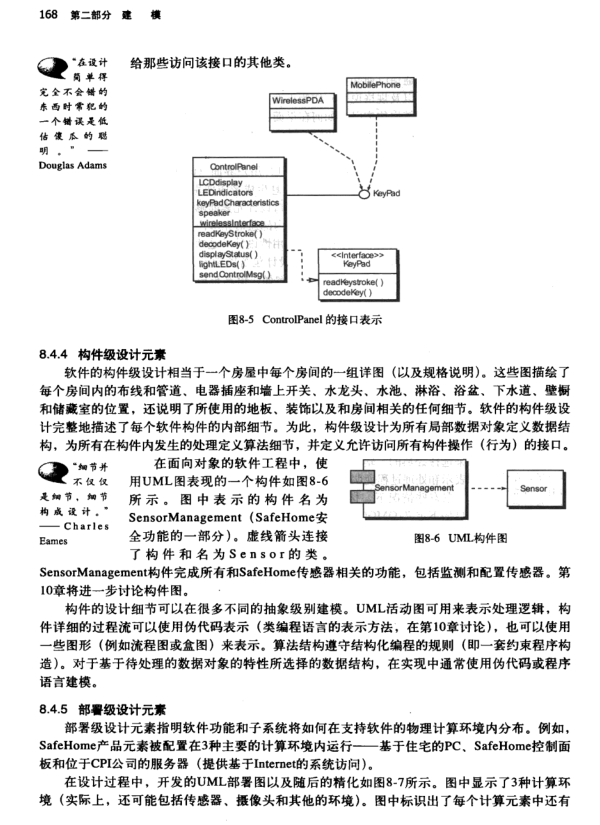
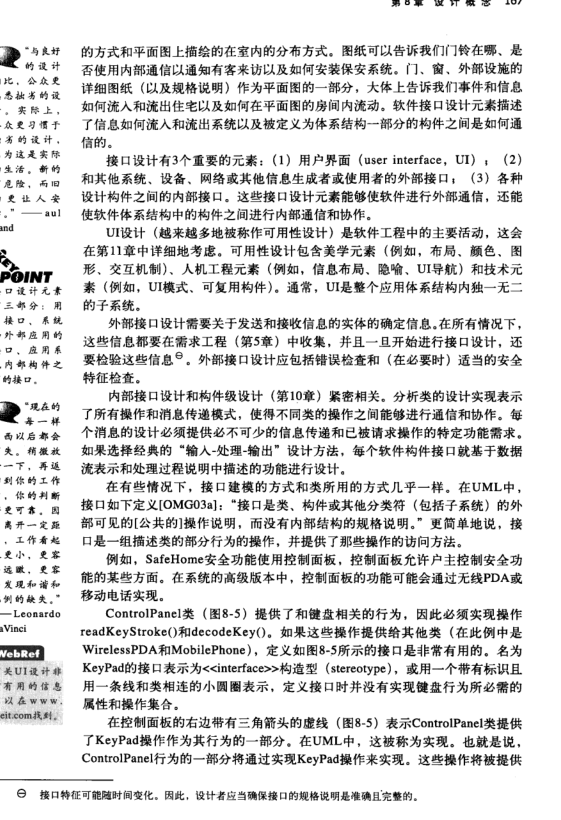
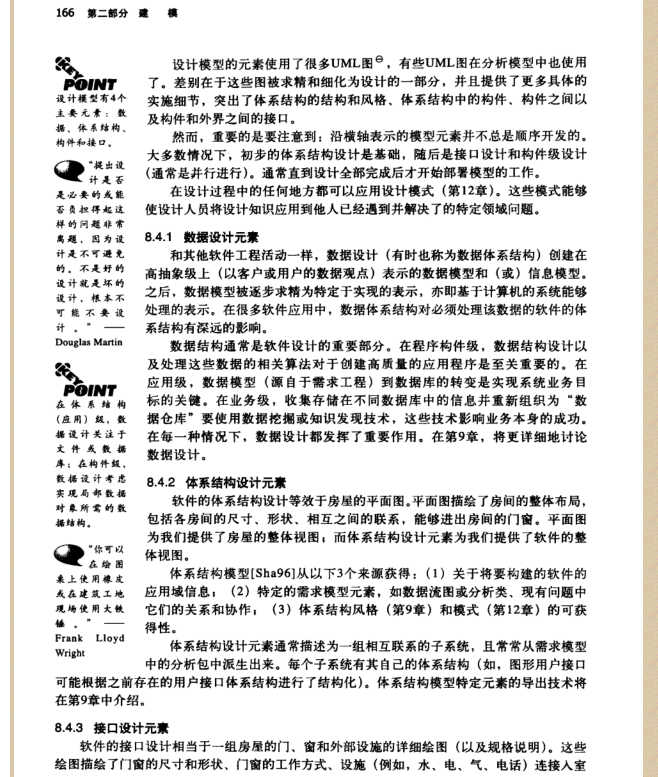
**（7）** **易维护性**

　　面向对象的抽象封装使对象信息隐藏在局部。当对象进行修改，或对象自身产生错误的时候，由此带来的影响仅仅局限在对象内部而不会波及其他对象乃至整个系统环境，这极大方便了软件设计、构造和运行过程中的检错、修改。

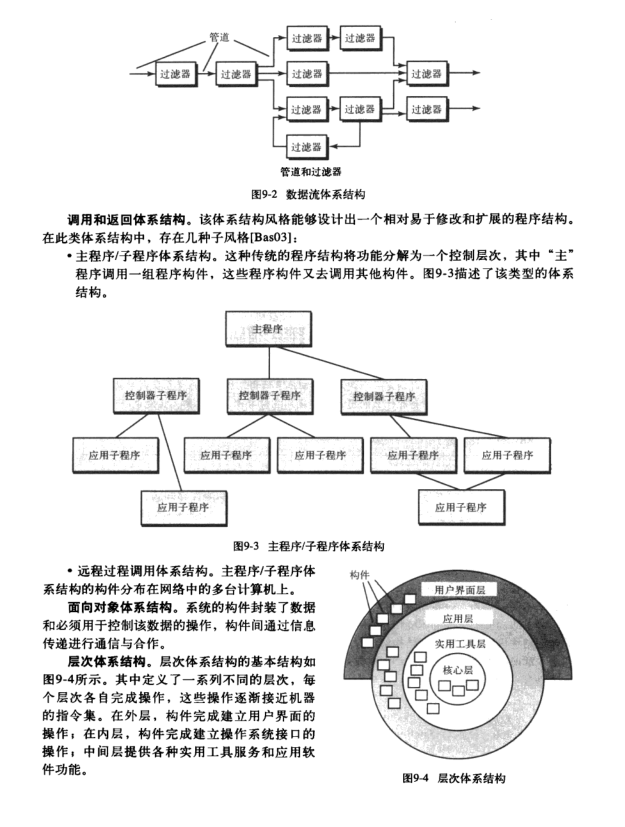
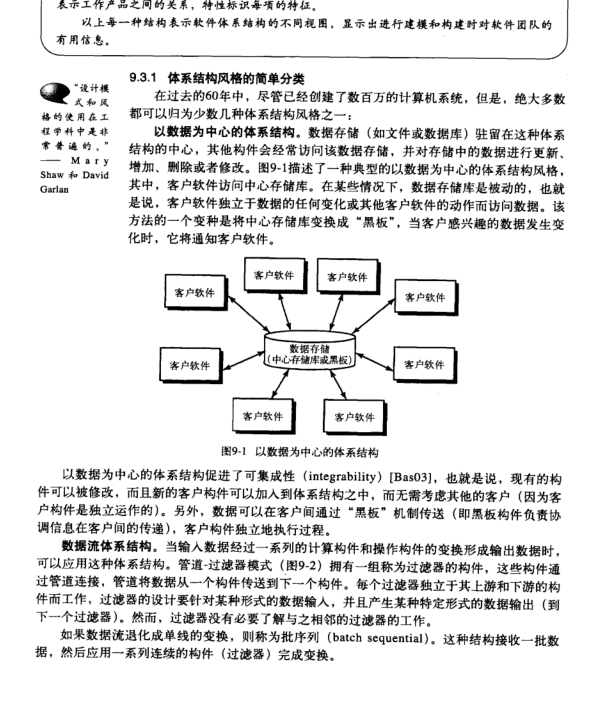
11软件设计要干什么



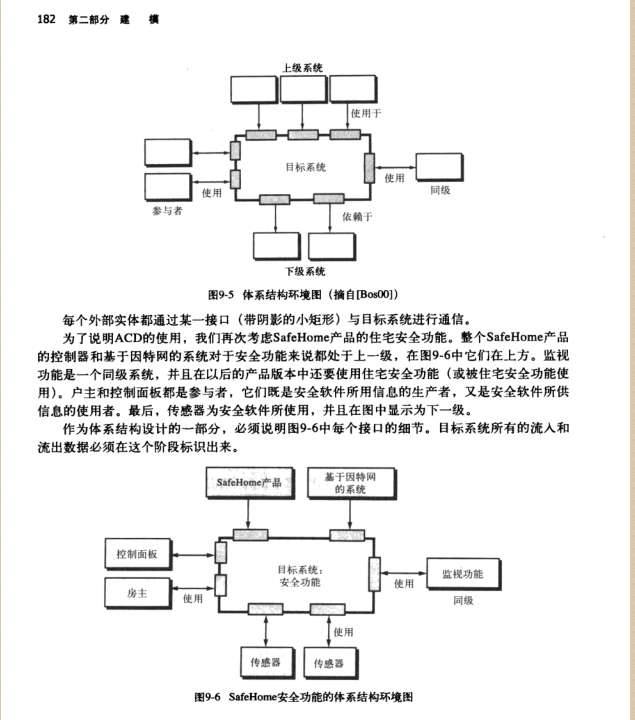
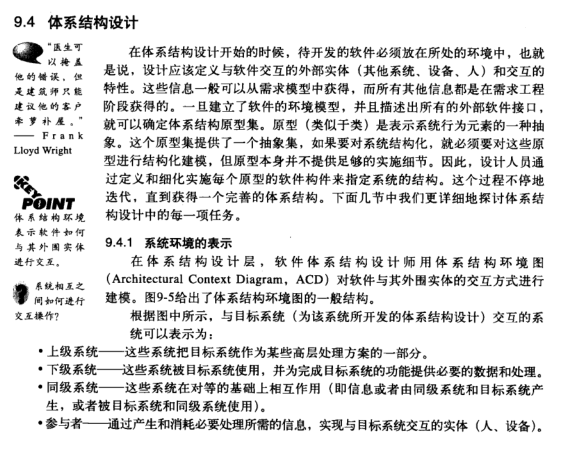
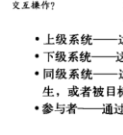
12软件设计模型有哪些元素



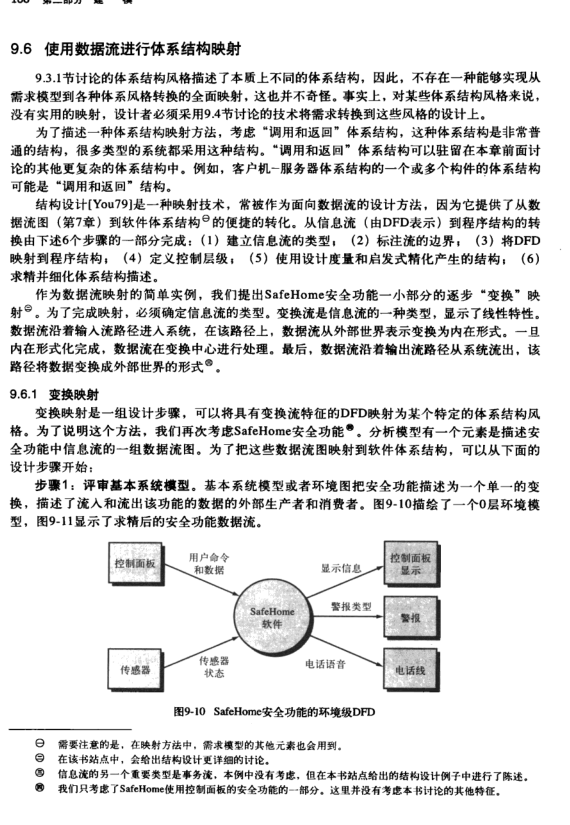
13体系结构种类



14体系结构在目标系统所属环境定义，有哪些元素。

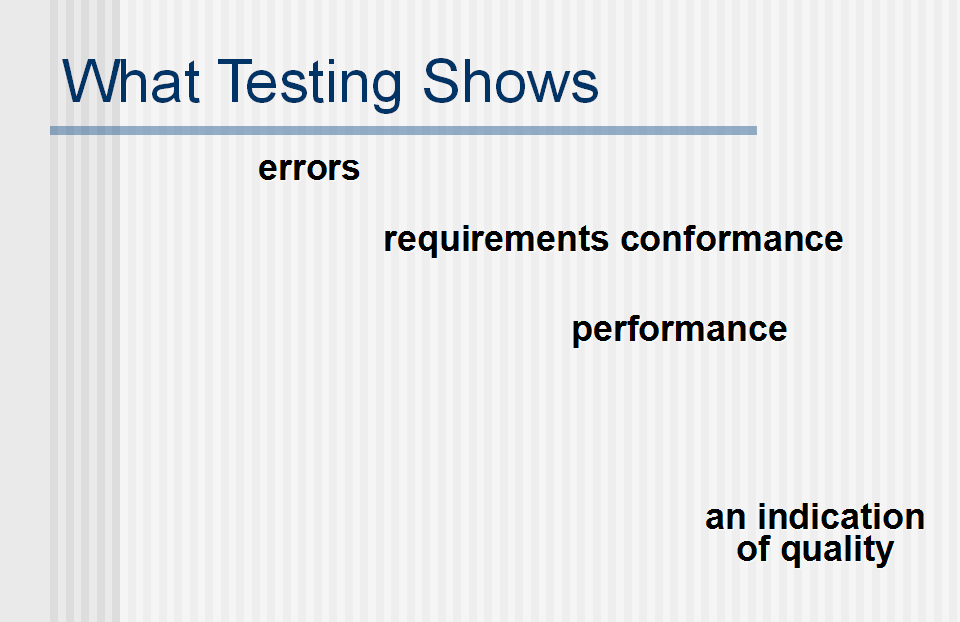


14数据流映射

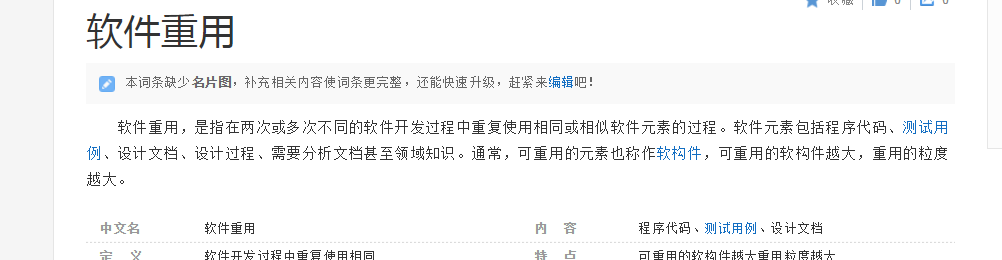


15软件测试目的，以及结果





16软件重用



17uml各种图的应用场合

