

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

# 系统展示层设计

Name: 汤涛

Job Title: 技术总监

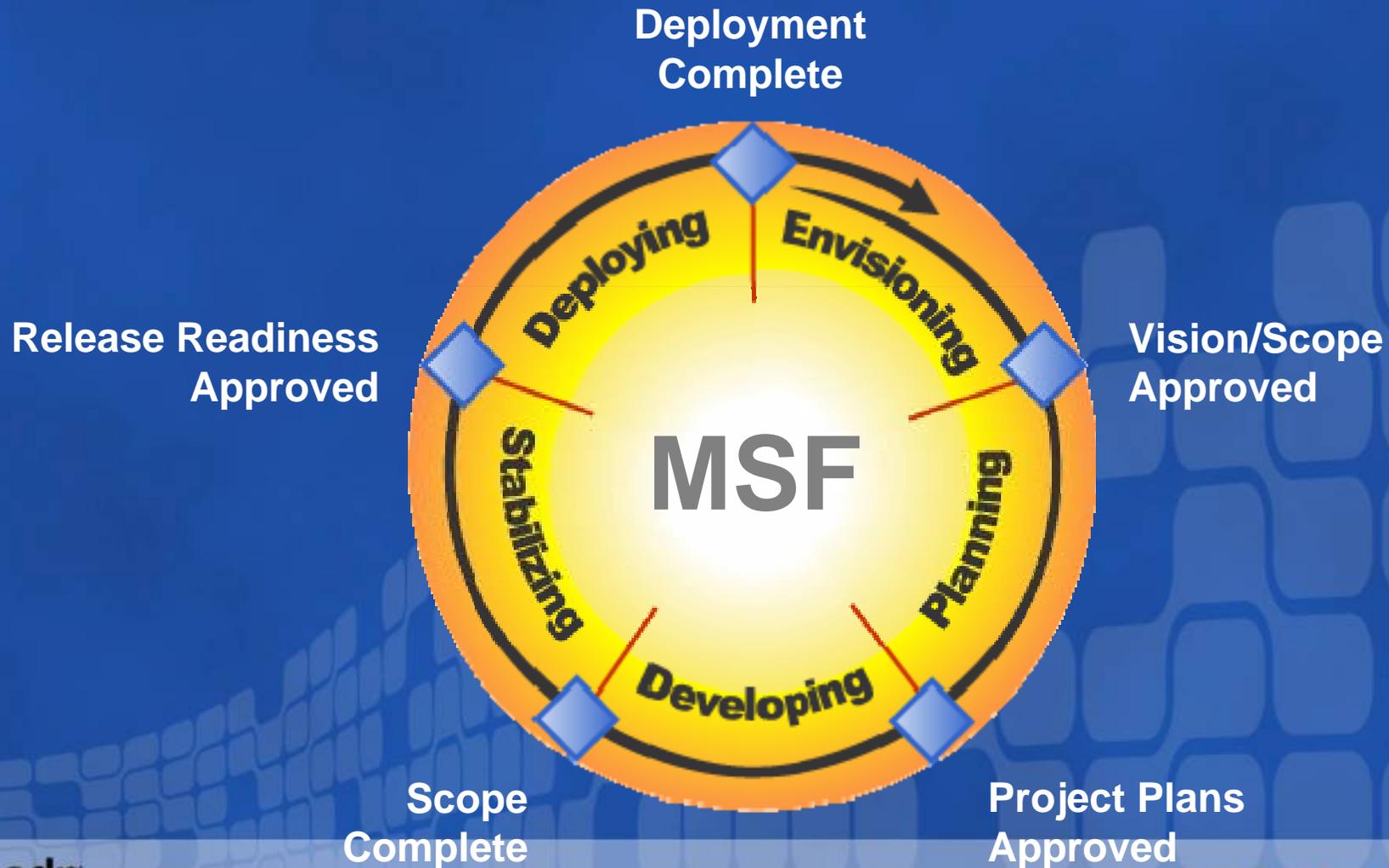
Company: 北京通嘉科技 ([www.onit.cn](http://www.onit.cn))

Email: tangtao@onit.cn

# 软件生命周期模型

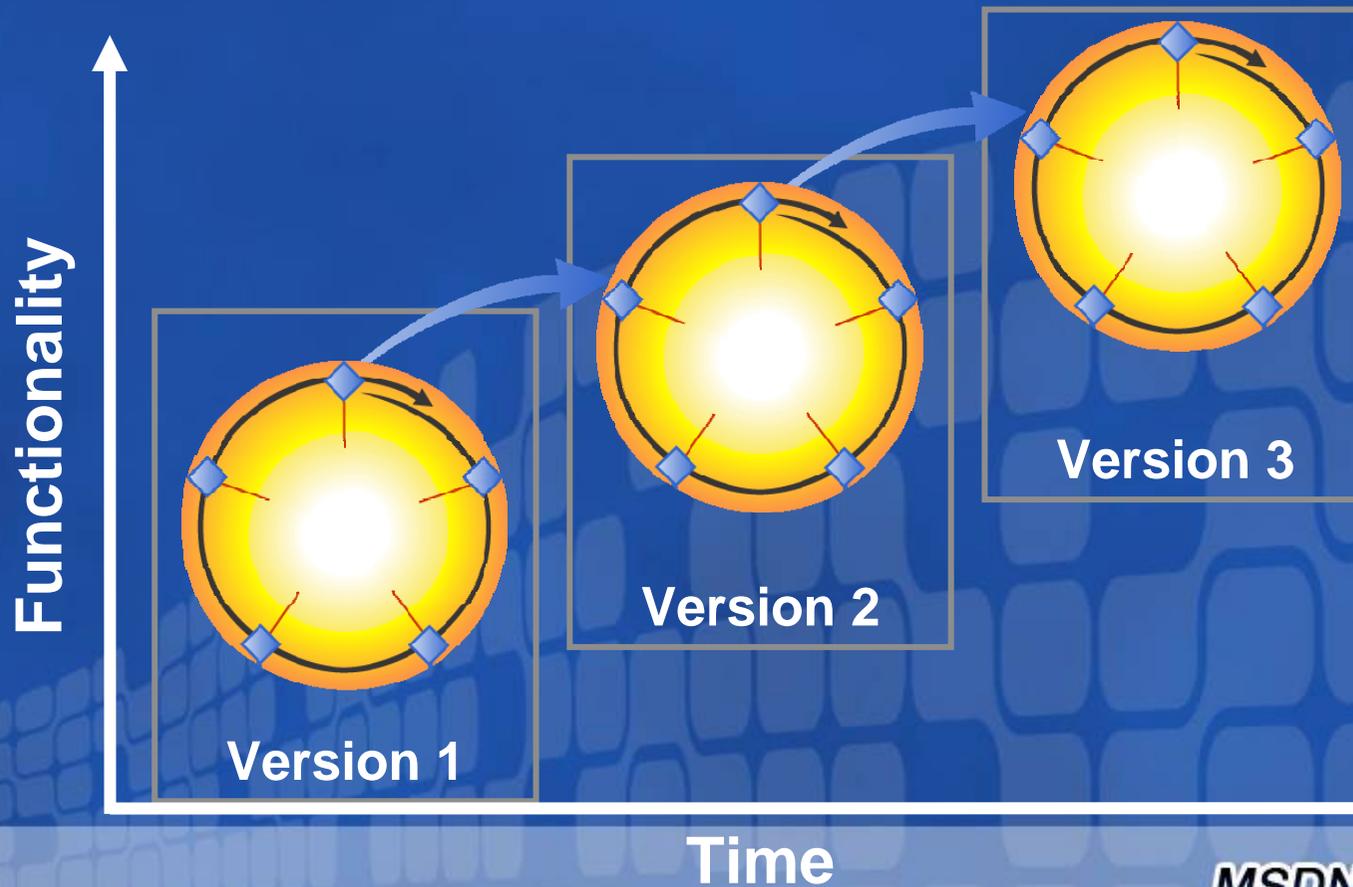
- 瀑布模型(Waterfall)
- V模型
- 编码修正模型(Code & Fix)
- 增量模型(Incremental)
- 渐进模型(Evolution)
- 螺旋模型(Spiral)

# MSF 过程模型

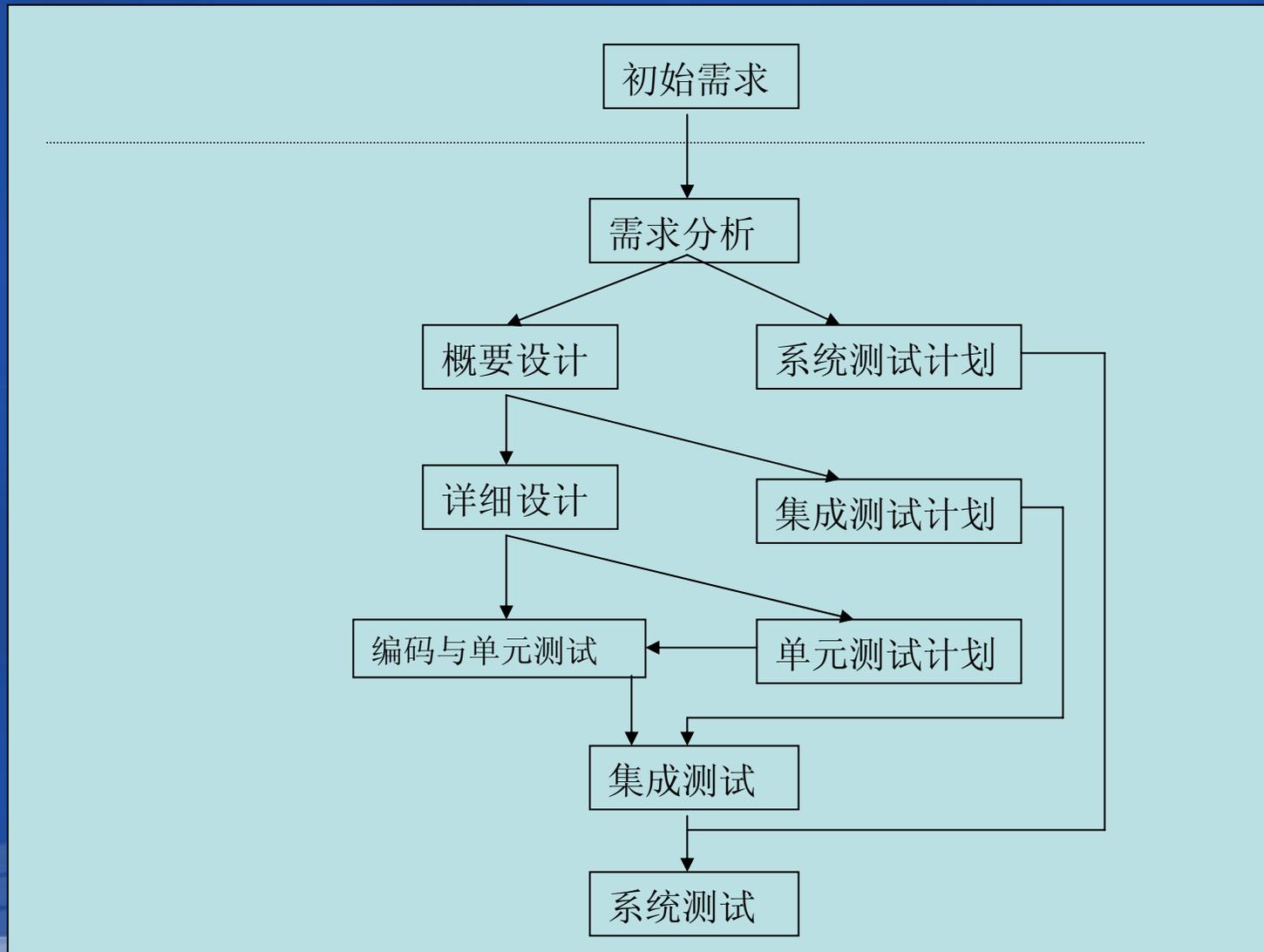


# MSF 过程模型是一种迭代方法

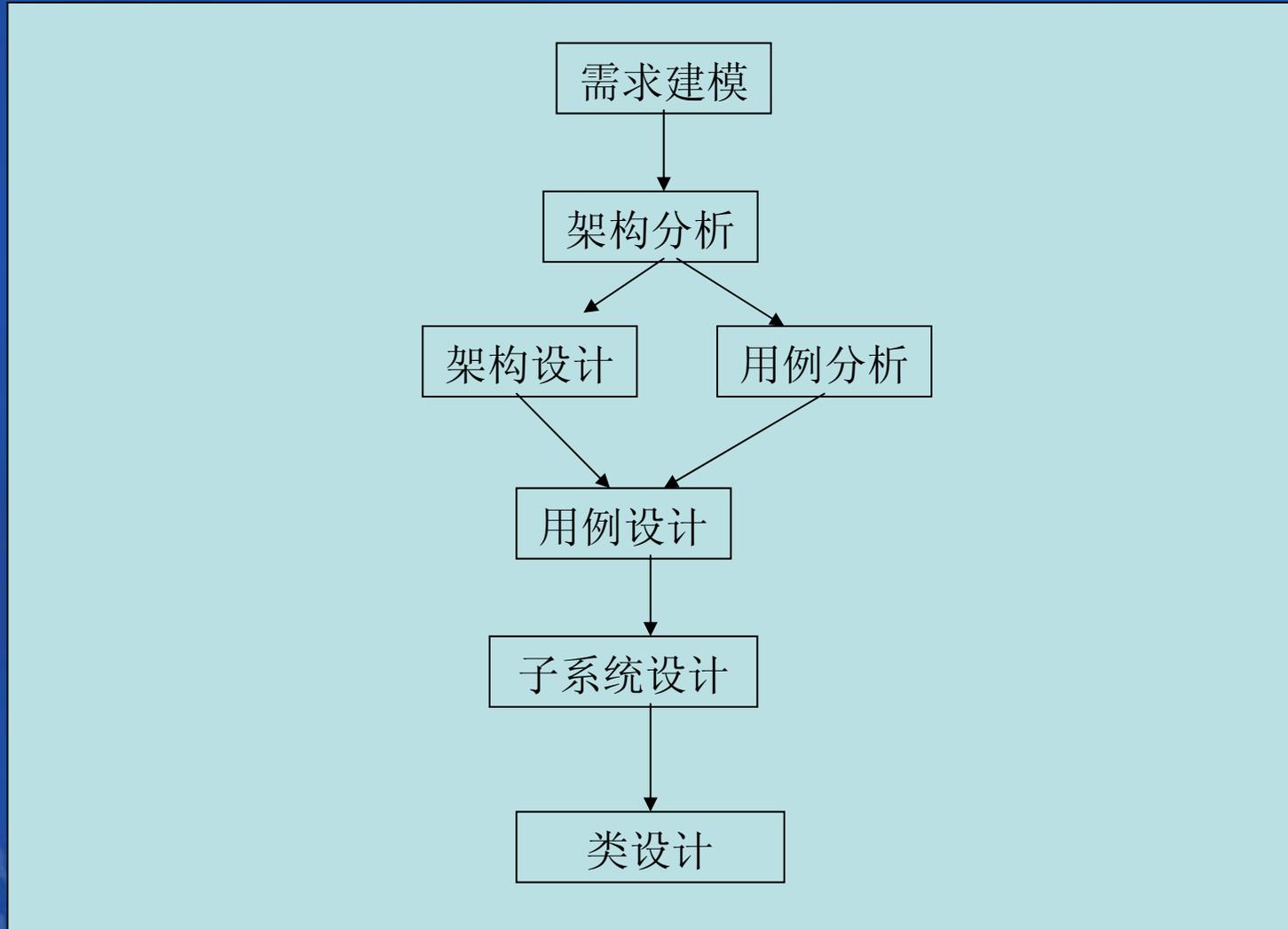
把大项目拆分成多个版本可以减小项目风险。



# 最常见的模型

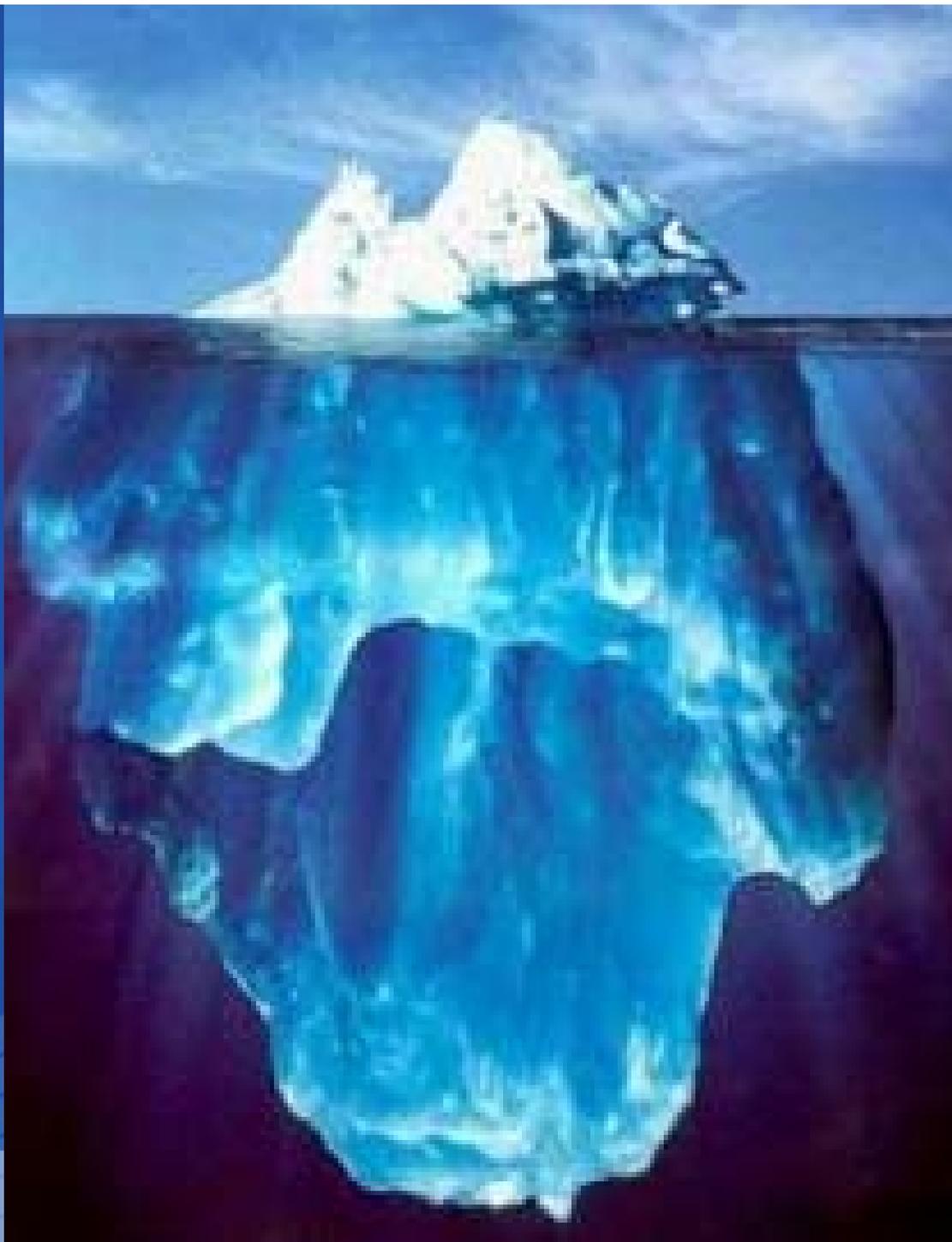


# 细化分析设计阶段



# 展示层设计概要

- 界面
- 功能
- 行为



您的潜力. 我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司



**MSDN Webcasts**

# 常见的设计

- 遵循标准的开发过程: MSF、RUP等
- 需求分析从功能出发
- 没有交互设计环节: 按照每个开发人员自己的体验和想法进行设计, 而不是按照用户的进行设计
- 忽视交互环境和用户习惯

# 开发人员和用户的区别

- 开发人员：
  - 实现方法
  - 逻辑、算法
  - 版本
  - ...
- 用户：
  - 交互
  - 感觉、体验
  - 价格
  - ...

# 界面和功能的统一

- 强调用户体验, 注重交互
- 采用交互设计方法

# 什么是交互设计

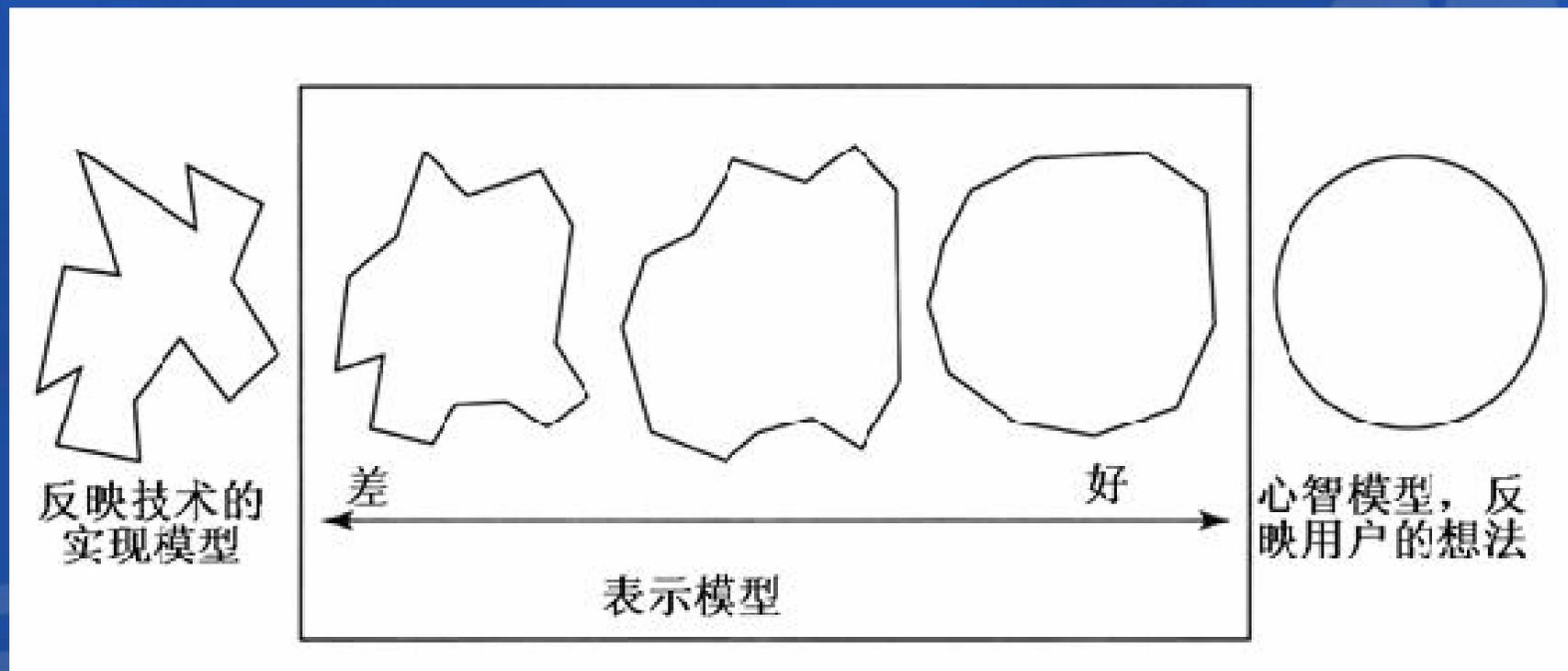
- 如何让产品易用、有效而且让用户愉悦的设计技术：
  - 了解用户的目标和期望
  - 了解用户的心理和行为特点
  - 定义产品的外观和行为
  - 建立人和产品的有机关系
- 1984年由比尔·莫格里奇提出

# 交互设计

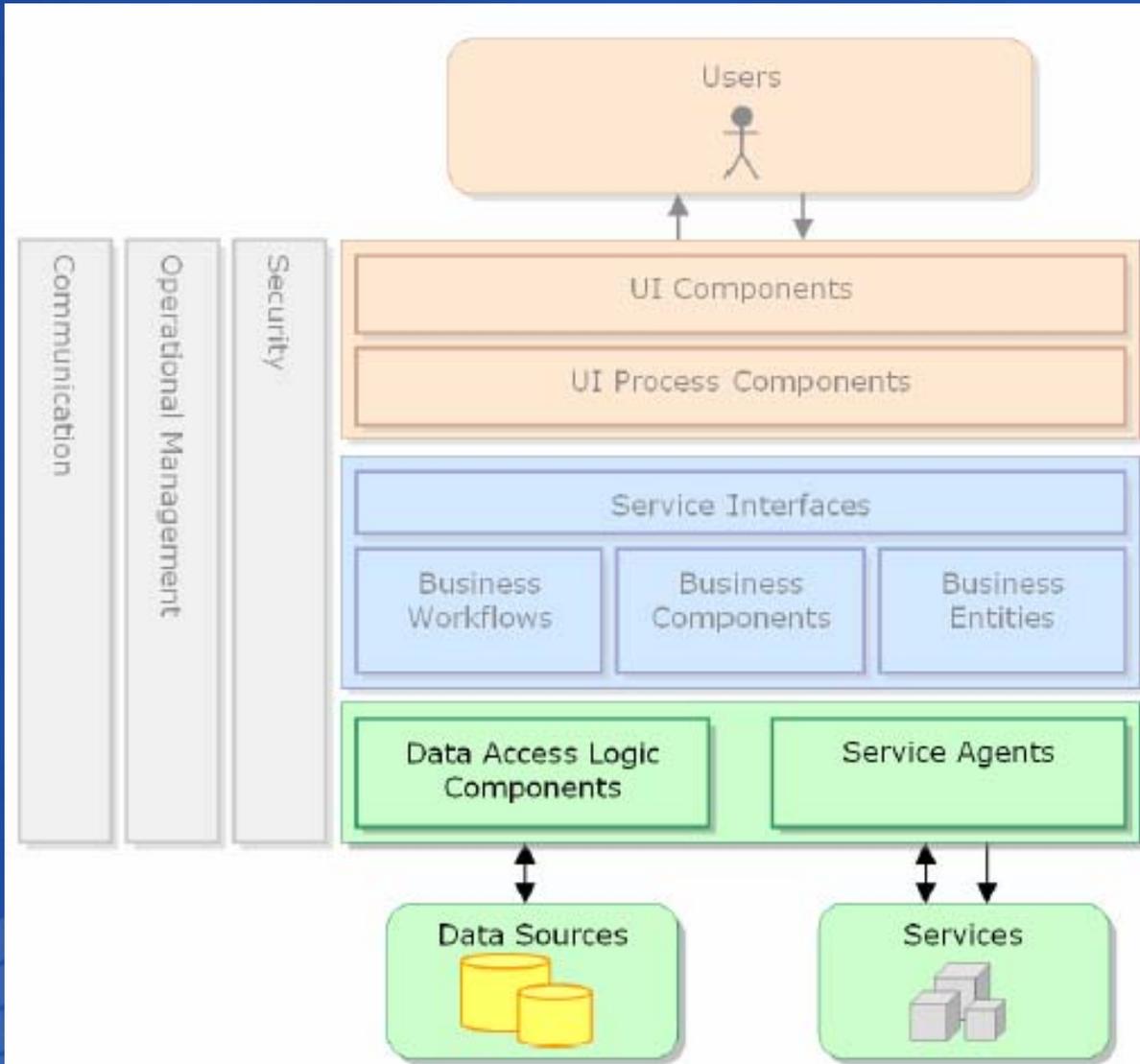
- 表达了软件外观和行为
- 帮助软件开发人员降低系统的复杂性
- 设置开发基线
- 提高开发效率
- 提高用户满意度
- 提高市场竞争力
- 提升企业品牌形象

# 设计过程-学习型设计

- 强调用户的感受和体验
- 确定用户的目标



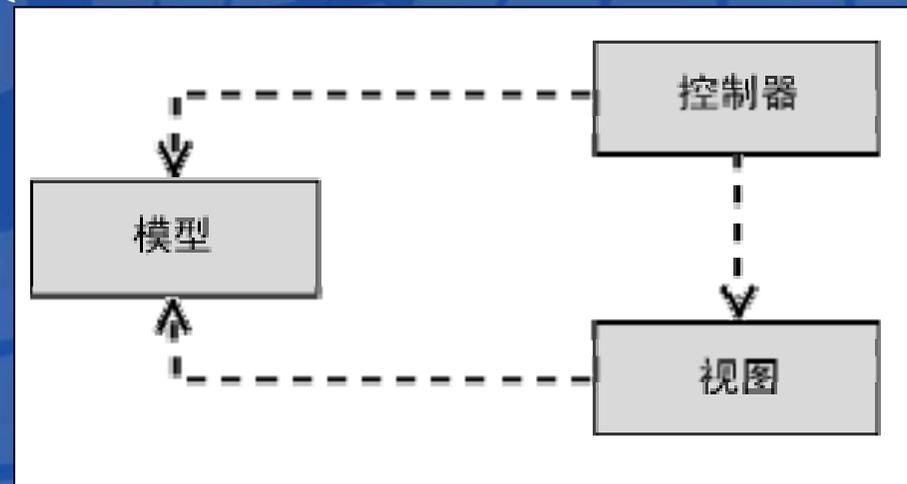
# 我们所讨论的问题



- 用户和应用进行交互的接口
  - 输入：
    - 辅助用户输入, 提供各种提示和帮助
    - 响应用户操作所触发的各种事件
    - 限制用户的输入
    - 处理一些特殊的操作, 如drag-drop, 剪贴板操作
    - ...
  - 展现：
    - 格式化数据
    - 特殊显示
    - 将一些代码翻译成可显示的名称
    - 其他(asp.net的页面cache、分页显示查询结果)
- 从UI的应用类型划分
  - Console、WinForm、WebForm、Plug-ins

# 与表现层的相关内容

- 如同一般的系统设计, 表现层也需要一整套支撑模块
  - 配置管理、Cache管理、本地存储、状态管理
- 表现层的设计模式



- 框架
  - 一整套能够动态配置和加载的插件模型
    - 启动时的加载项、动态的菜单项、工具栏, UIP组件...
    - 提供一个Controller, 能够管理UI与UI之间的调用
  - 一整套为各程序模块所使用的工具
    - 上下文对象: 用户信息、权限信息
    - 访问Service的公共入口: 集成认证
    - 客户端的Cache: 业务数据、MetaData、用户凭证
    - 客户端的应用配置信息
    - 客户端的本地存储机制
    - UI的scheme
    - 异常处理机制和最终的异常处理点
    - ...
- 应用程序的入口和框架之间关系
  - 浏览器: 可能是第一个引用的Control(Assembly)
  - WinForm应用: 登录窗口、主窗口、Splash窗口、TrayIcon
- 选择你的窗口应用类型: Dialog、SDI、MDI

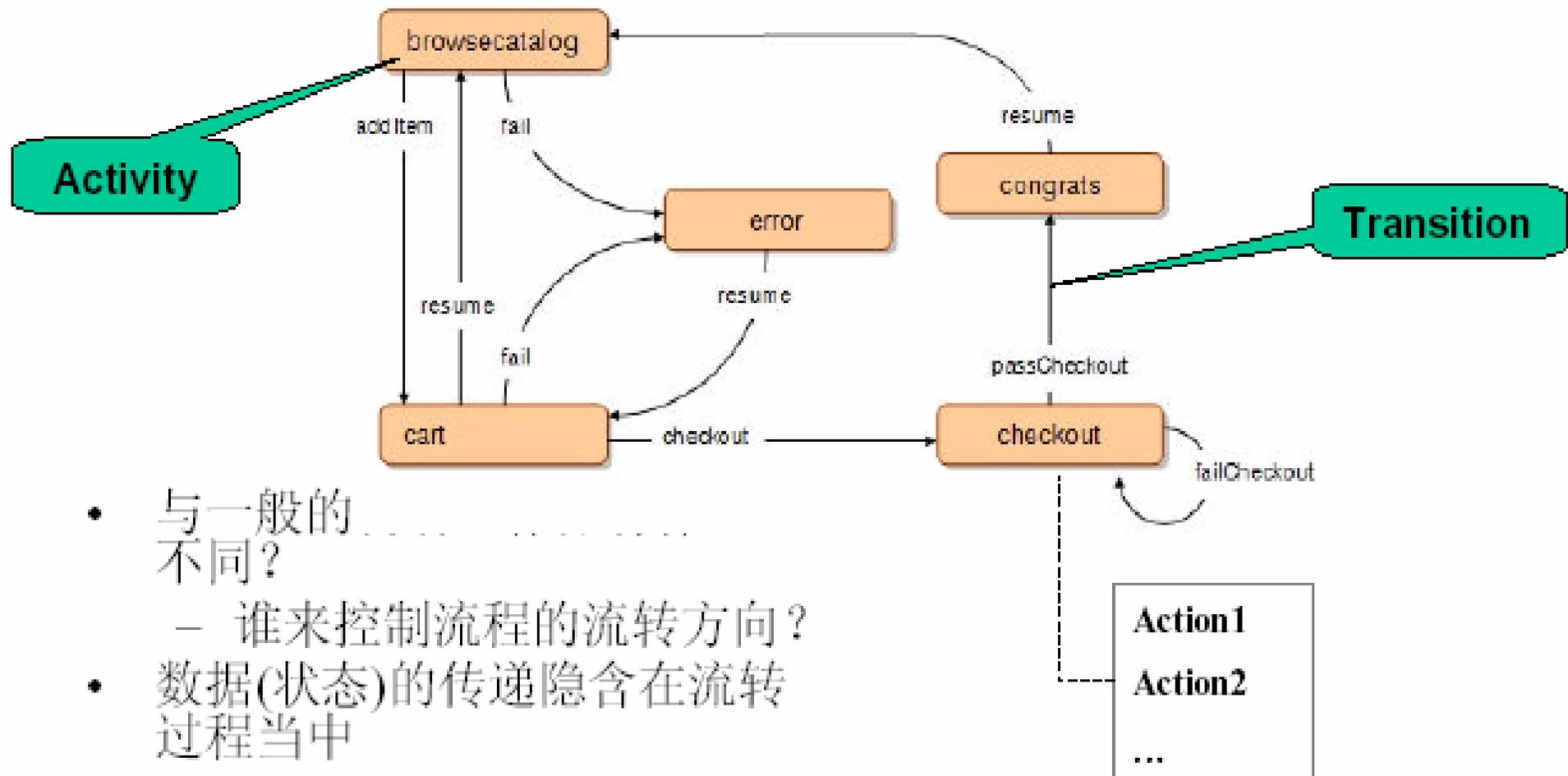
# UI与UI之间的调用

- WinForm应用
  - –Form与Form之间、Form与控制之间、控制与控制之间
- 浏览器应用
  - –页面与页面之间、页面与其它浏览器之间、页面中的不同部分之间
- 尽可能不要将UI与UI之间调用的代码写在UIComponents的代码当中
  - –由专门的Controller来完成UI的加载
  - –尽可能保证UI Components的重用性

- 根据状态的改变决定使用哪一个UI
- 应用场景
  - 有些UI之间的相互作用时，存在明确的处理顺序
    - 一个向导型的界面。用户可以选择上一步、下一步或者束
    - 在一个购书网站，用户可以反复地浏览上架的图书，然后将选中的图书放进购物车。最终会走到收银台结帐
  - 这些类型的界面操作的特点
    - 用户的操作流程可以用一张流程（导航）图来描述
    - 导航图上每一个节点就是一个用户界面（窗口/页面）
    - 界面之间的跳转是由用户操作触发的
- 处理这种流程的控制器，我们称为User Interface Process Components
  - 隔离了UI与业务逻辑层
  - 对流程中的UI进行了管理
  - 提供了状态保存和传递的机制

# 购书流程图

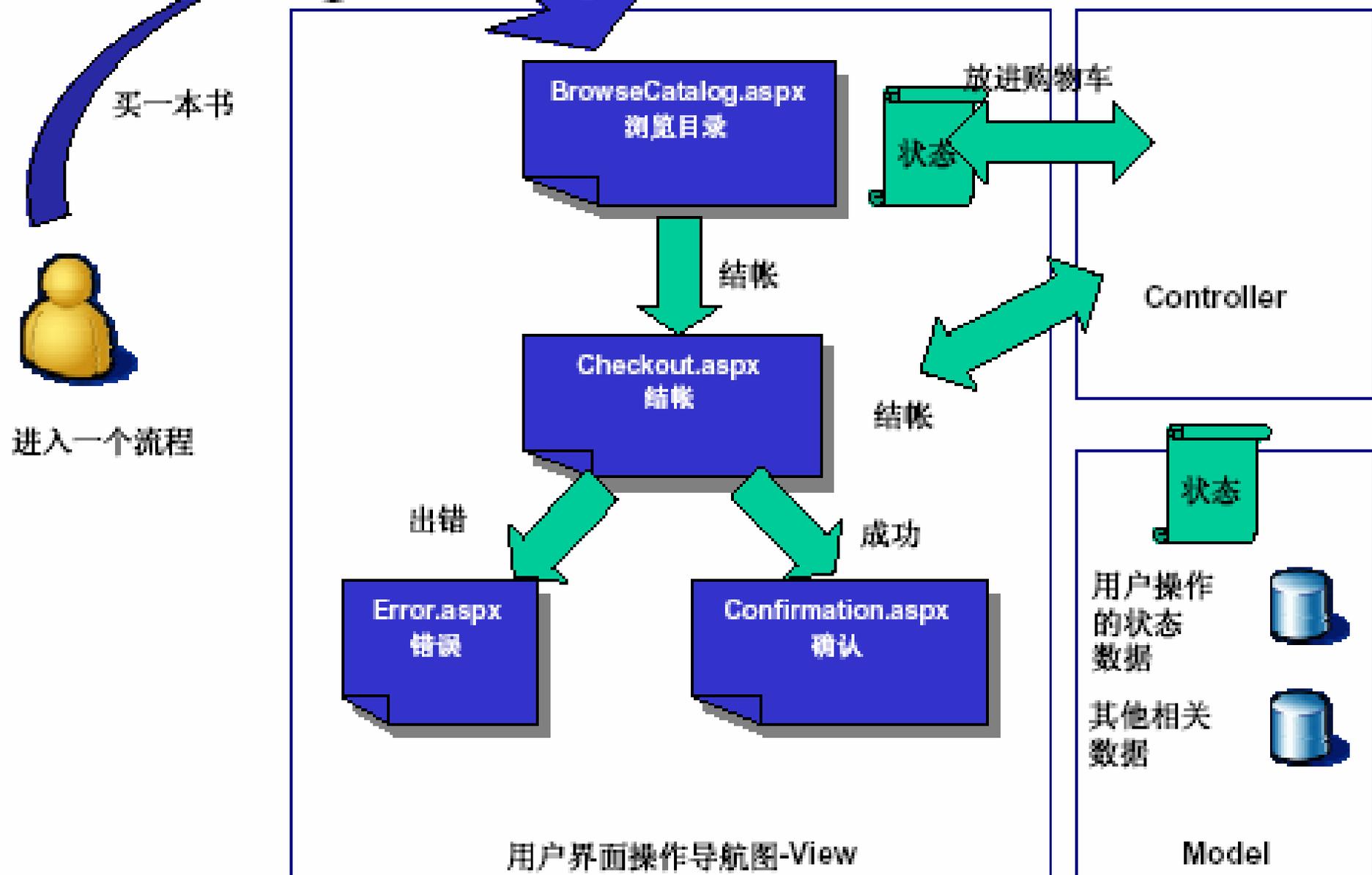
- 网上购书的流程图



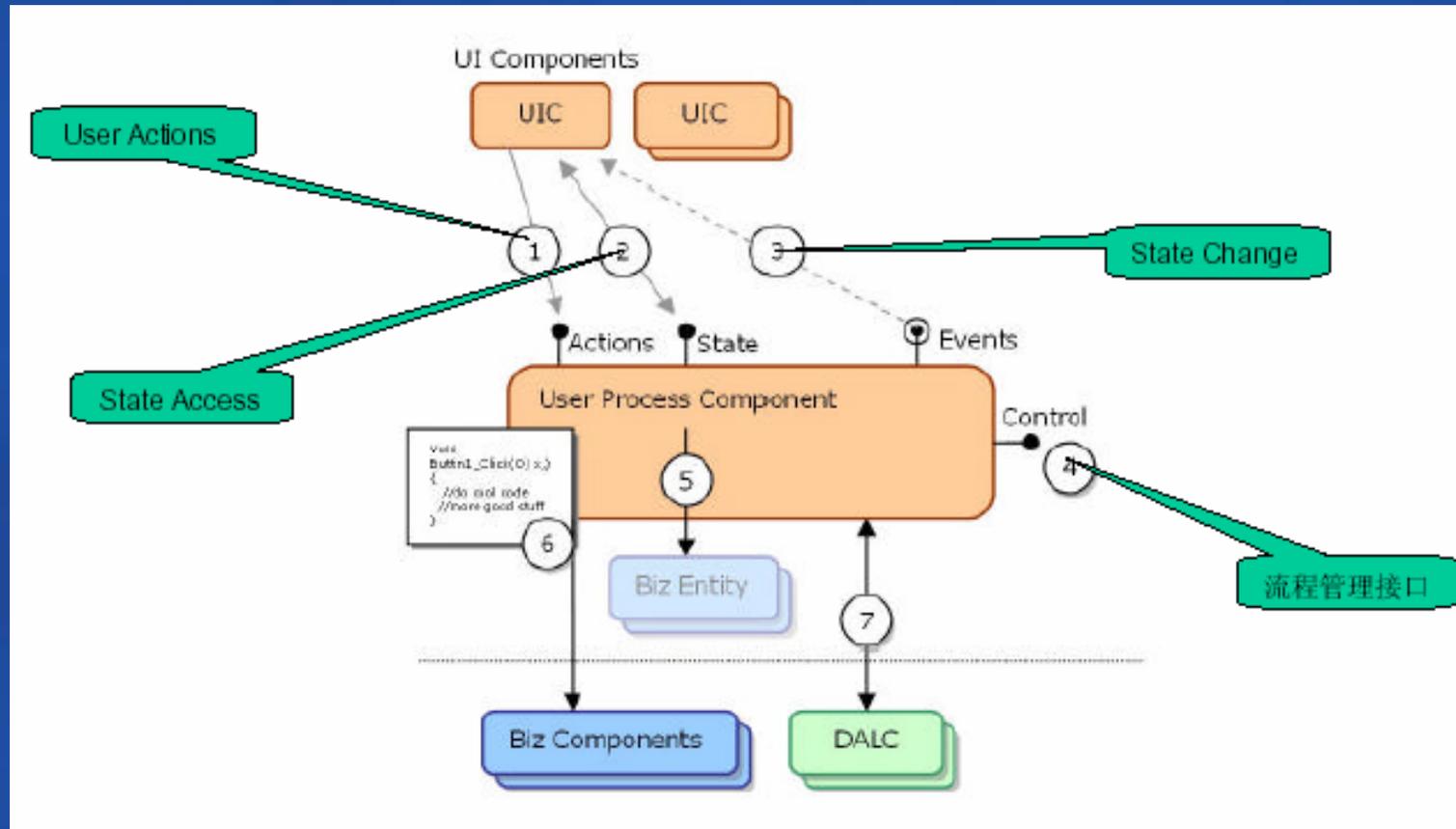
- 与一般的...不同?
  - 谁来控制流程的流转方向?
- 数据(状态)的传递隐含在流转过程当中

# Microsoft User Interface Process

## Building Block-View Controller和State



# UIP接口设计



# UIP Components

## 需要考虑的问题

- 其他需要考虑的问题
  - 如何将控制流程和窗口、页面等界面元素分开
  - 流程的每一步都需要保存状态。如何能抽象出一个统一的状态模型, 可以使状态在留转过程中被保存和传递
  - 最好能够用一套处理机制, 能够满足inForm程序和Web应用的需要
  - 用户的在一个操作流程中可能会嵌套子流程, 或者会转到别的流程
  - 记录流程的流转的日志
  - 流程的维护
    - 流程改变对正在运行的流程的影响
    - 能否通过某种特殊手段改变流程的状态?

# Microsoft User Interface ProcessBuilding Block

- 为解决上述问题而提供的一组接口和类
- User Interface Process Application Block(UIPB) 用一种XML格式来表现流程图
- 在UIPB中, 会用一些接口/类来表示图中的内容
  - ITask, 用来表示整个流程
  - IView, 用来表示图中的(节点)界面
  - State, 表示界面的状态。例如购物车中的书名和数量
  - ControllerBase, 控制界面之间的跳转、控制Task的启动和结

## ProcessBuilding Block的设计模式

- UIPB采用了MVC模式
  - Views, 任何实现了IView接口的类(Form或WebForm)
  - Model, State类, 它保存的是界面上的信息以及控制信息
  - Controller, 就是ControllerBase及其派生类

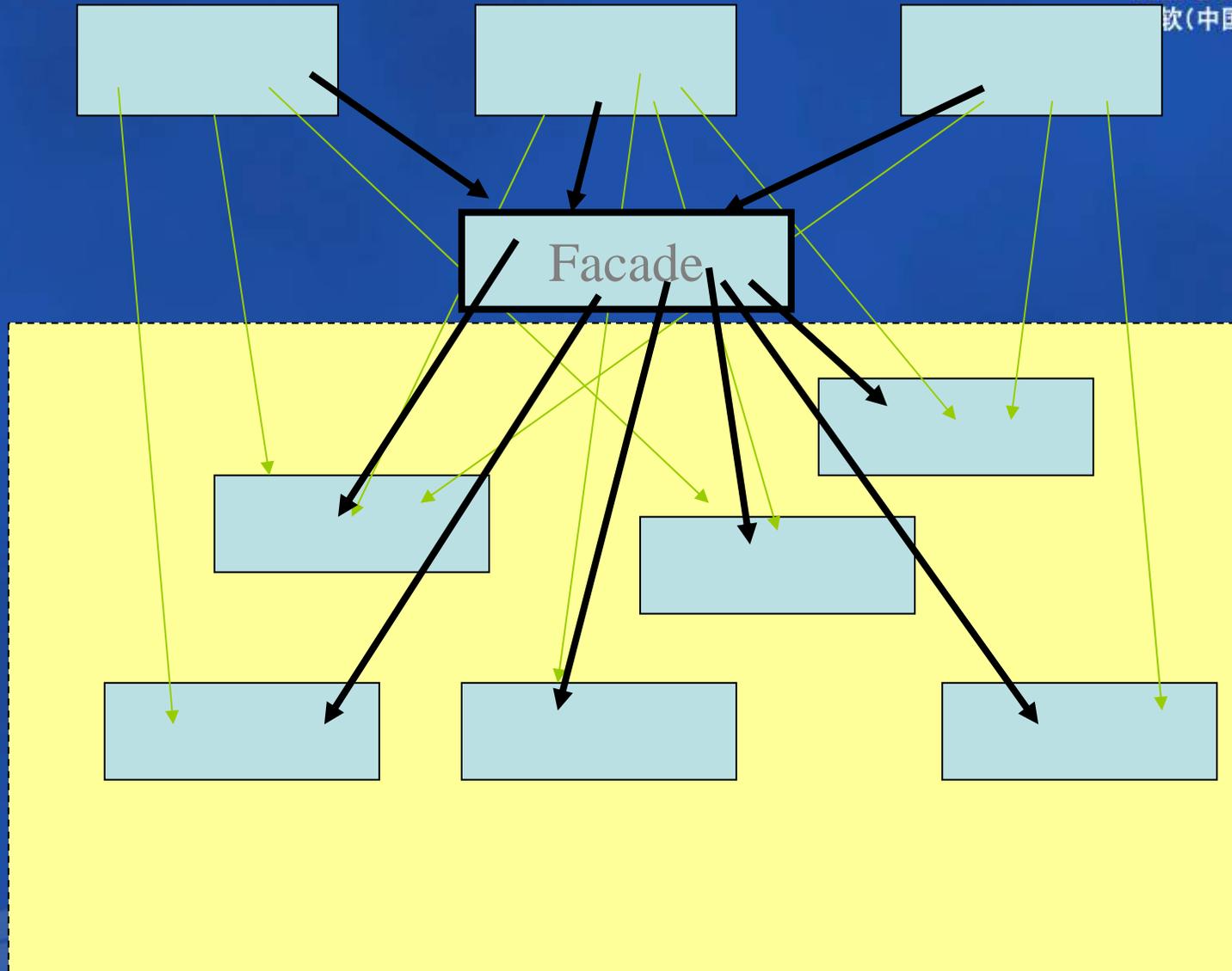
## ProcessBuilding Block的其它功能

- 保存状态
  - 定义了 **IStatePersistence** 接口，应用可以自己定制自己的状态保存方式
- WinFormView, WebFormView
  - 实现了 **IView** 接口。
- ViewManager
  - 管理在流程中出现的 View
- UIP Config
  - 读取应用程序配置文件信息，根据配置文件构造相应对象和流程图
- UIP Manager
  - UIP 的入口，启动一个新的流程或已知的流程

DemoTask task = new DemoTask(); //实现了ITask接口的定制流程

# UIP设计考虑

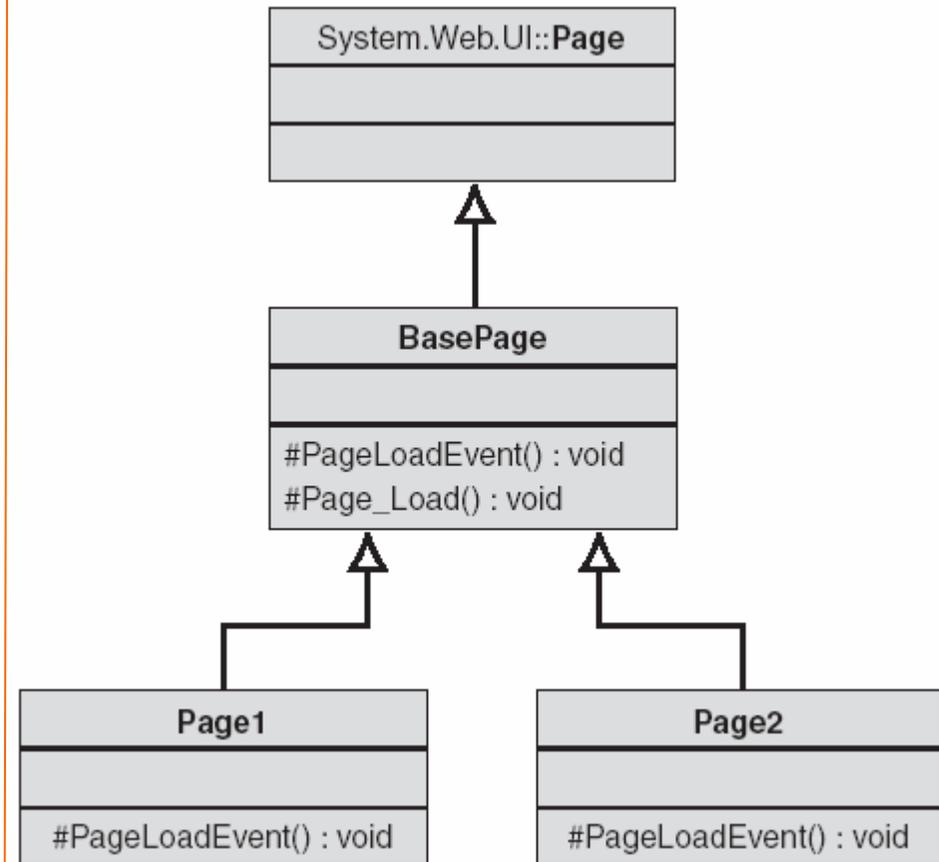
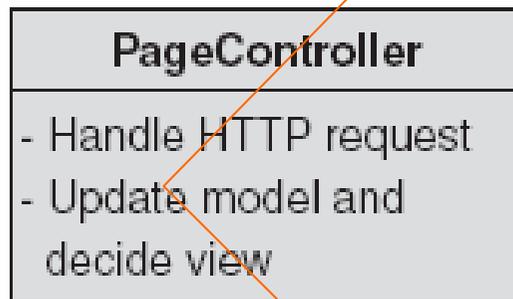
- 作为设计者, 应该充分考虑到UI模型的复杂度
- 在UI与业务层之间, 可以增加UIP层进行隔离
  - UIP管理了什么
    - 流程、状态、View...
  - UIP的Building Block仅是一个模型, 还需要通过与实际业务逻辑结合使用。



# 页面控制器

- 模式名称: Page Controller
- 要解决的问题: 在MVC中如何为复杂的Web应用程序组织控制器以便在避免重复代码地情况下实现重用性和灵活性。
- 解决方案: 使用页面控制器模式接收页面请求, 然后在模型上调用请求的操作, 以决定用于显示页面的正确视图。

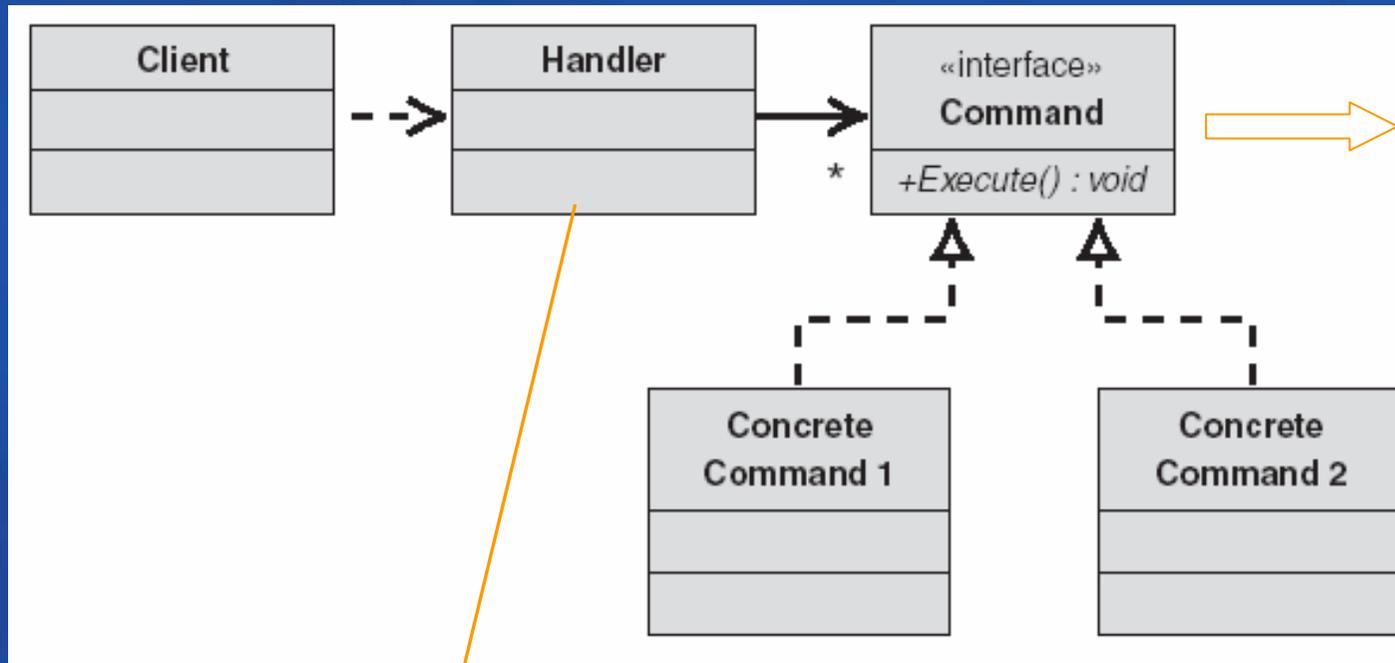
# ASP.NET实现页面控制器



# 前端控制器

- 模式名称: **Front Controller**
- 要解决的问题: 在MVC中如何为复杂的Web应用程序组织控制器以便在避免重复代码地情况下实现重用性和灵活性。
- 解决方案: 处理器 (**Handler**) 接收HTTP的**POST/GET**请求后, 检索相关请求参数, 以选择正确的命令, 并传递给具体的页面做相应的命令处理。

# ASP.NET实现前端控制器



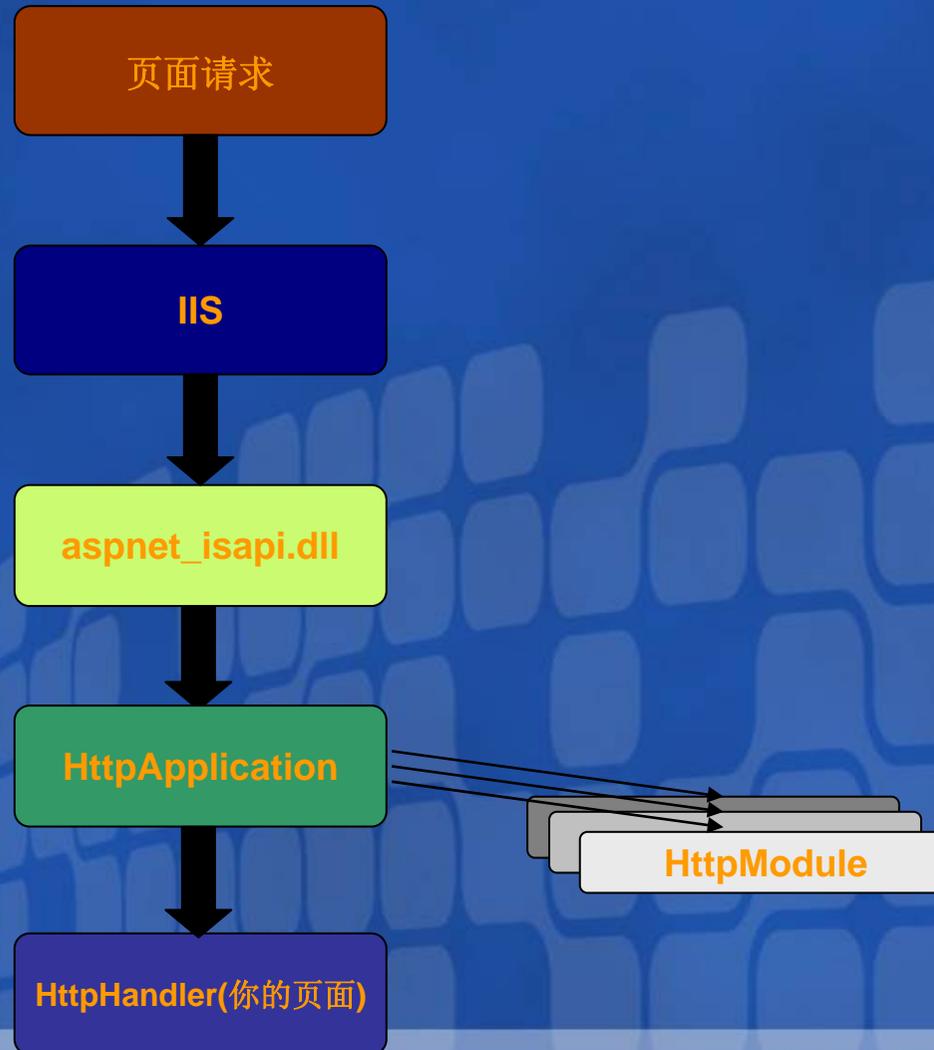
页面处理

IHandler

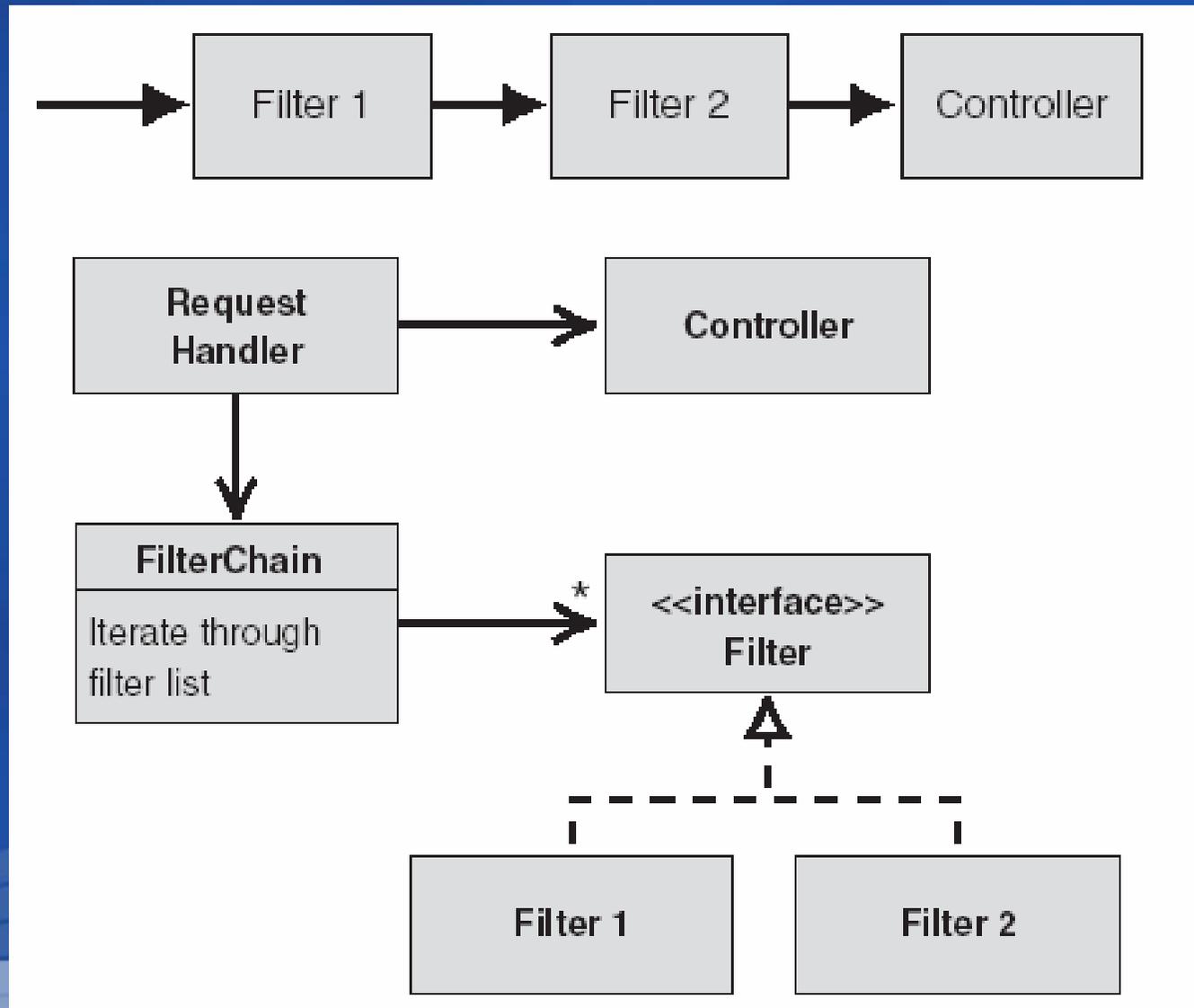
# 拦截过滤器

- 模式名称: **Intercepting Filter**
- 要解决的问题: 在Web请求过程中如何实现通用的处理前和处理后操作?
- 解决方案: 创建一组链式过滤器, 实现在Web请求过程中通用的处理前和处理后操作。

# ASP.Net的Request处理过程

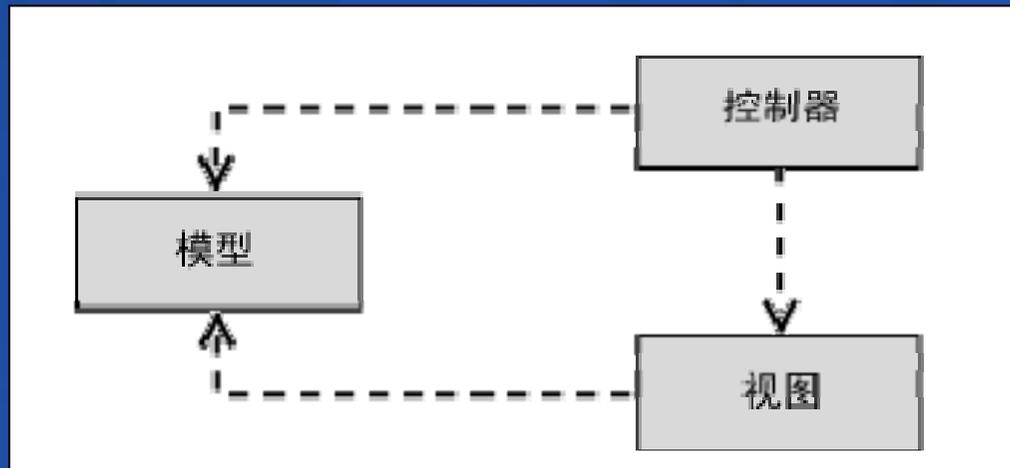


# 原理



# ASP.NET处理客户端 请求的模式

- ASP.NET中通过页面与代码的分离, 实现了MVC模式



- 其中页面中的HTML以及代码中生成HTML的部分构成了视图(View)
- Page\_Load和Button的Click事件实际上是控制器(Controller, 更具体一些是Page Controller)
- 而数据访问部分可以看作是模型(Model)
- 优点、缺点?

# ONIT 通嘉科技

首页 服务 培训 产品 论坛 关于

软件开发咨询

我们为软件企业提供如下

- 开发咨询服务
- 项目团队培养
- 系统架构设计

企业技术培训

- 软件架构师
- 项目管理
- 配置管理
- 需求工程
- 分析设计
- UML
- 设计模式
- 软件测试
- .NET开发
- J2EE开发
- 数据库设计
- 数据仓库

## 欢迎访问 www.onit.cn



以提供现场解决问题的技术咨询  
服务,也可以通过电话、Email和  
即时通讯等方式来远程解决问  
题。通过这种方式能在最短时间

解决项目实施过程中的关键问题,扫除技术障碍,为您的项目  
按期高质量完成提供坚实雄厚的保障。

**项目团队培养:**古人云“磨刀不误砍柴功”。对项目开发团  
队的前期培养是保证项目进度的一个重要手段,我们采用  
大型项目实战场景,为您的团队成员快速进入项目角色提  
供富有成效的指引,为他们提供丰富的项目经验,让项目  
团队成员在将来实际的项目开发过程中有效规避技术和管  
理陷阱,充分发挥出每个成员的各方面潜力,用其所长而  
避其所短,最终节省项目成本、缩短产品上市时间、提高  
客户满意度等等。



优秀的软件系统架构设计将为  
后续的开发提供良好的基础。

我们所提供的企  
业技术培训是基  
于实践的培训,能  
让您在经过培训后  
立即将所学的知识  
用于实际工作。我  
们的讲师不仅具有  
丰富的实践经验,

而且也具备丰富的教学经验,是中国教育  
培训界技术水平和教学水平一流的讲师。  
目前教育培训界很多优秀的讲师都曾从师  
于我们的讲师或从师于我们讲师的学生。

**软件架构师培训**是基于目前国内紧缺的  
高端软件人才而开发的一整套课程体系,  
其目的就是为了让您在经过培训之后,能  
快速地加入到中国软件架构师队伍中来,  
为加快国内软件业的发展助上一臂之力。  
该课程将以目前国内大型的软件项目为案  
例,让您深入了解架构设计的各个环节,  
是您进一步深造的必修课。

**设计模式 (Design Pattern)** 是软件设计  
过程中非常重要地方法和策略,可以更为  
有效地重用设计模型和代码,提高设计和  
开发效率。我们提供专门针对设计模式的  
培训,让您在短时间内把握C#的22种

# 获取更多MSDN资源

- **MSDN中文网站**  
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**  
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**  
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**  
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.mspx>

# Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

The screenshot shows a window titled "问题和解答 (无问题)" (Questions and Answers (No Questions)). The window contains a text area with the message "在此会议中尚未解答任何问题。" (No questions have been answered in this meeting yet.). Below the text area are three buttons: "提问(A)" (Ask Question), "删除(D)" (Delete), and "问题管理器(Q)" (Question Manager). A text input field at the bottom left contains the placeholder text "要向演示者提问，请在此处键入问" (To ask a question of the presenter, enter your question here).

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

**Microsoft**<sup>®</sup>

msdn  


**MSDN Webcasts**