

基于敏捷技术的敏捷开发 辨析与应用研究

戴洪蕾, 王建国

(内蒙古工业大学, 内蒙古 呼和浩特 010051)

[摘 要] 高效便捷的轻量型软件开发模式是管理信息系统开发的新方法和新思路, 作为敏捷开发模式基石的面向对象技术为这一方法提供了可行性。简单三层将系统逻辑分为三层保证了系统的安全性和扩展性, 配合抽象工厂模式和 .Net 反射机制, 以及时下新兴的 Web 开发技术 ASP. Net MVC 的优势, 使得敏捷开发模式在实施中变得更加高效。本文以敏捷开发模式为基础, 研究如何将简单三层、抽象工厂模式和 ASP. Net MVC 等技术相结合, 从敏捷技术角度完善敏捷开发模式。

[关键词] 敏捷开发; 抽象工厂模式; ASP. Net MVC

[DOI] 10.13939/j.cnki.zgsc.2015.21.053

伴随着世界经济和技术的大力发展, 计算机技术得到了更加广泛的应用, 越来越多的工作需要依赖计算机的智能和高效。新的软件的出现都伴随着新的软件开发模式的出现, 新的软件开发模式出现的原因就是解决在固定模式中出现多变的需求。而传统的开发模式存在着面对需求变化困难, 维护和扩展性差导致系统的二次开发成本较高, 敏捷开发模式应运而生。因为人们的需求各种各样, 为了使人们的需求得到解决, 软件技术也在进行不断更新完善, 所以敏捷技术的产生也为敏捷开发模式的实施提供了有效的保证和可行性。

1 敏捷开发

敏捷开发是由 15 个科学家共同提出来的, 其中包括来自思特沃克公司 (Thought Works) 著名的软件大师马丁·福勒 (Martin Fowler)^[1]。敏捷开发即一种全新而快捷的软件开发模式, 是把人放在第一位, 以满足用户不同需求为导向的开发模式^[2]。应用敏捷开发的方法, 要求团队成员具有很强的主动性, 满足了高内聚、松耦合的原则把项目分成了若干小组。较少的文档准备和分组的新方式缩短了软件开发周期, 开发过程中的多次迭代和测试提高了软件的质量^[3]。

2 敏捷技术的重要性

从目前敏捷开发的研究现状中, 我们可以看到在任何一个软件的开发过程中, 任何一种模式都不是能解决所有问题的万能钥匙, 所以在软件开发过程中应全面考虑所有的问题, 不管是开发方法、设计模式还是设计架构都是必须考虑的重要因素, 而就目前敏捷开发的应用研究来看, 敏捷开发的过程, 对技术并没有明确的指导思想, 这也正是多数项目在应用敏捷开发过程中失败的重要因素。

从根本来说, 敏捷开发不仅是一个软件开发过程的方

法论, 准确地说它更是一种思想, 但这种方法是在敏捷技术上的, 敏捷技术为这一方法的实施提供了可行性, 只有敏捷的技术才能支撑敏捷开发的实施^[4]。真正的敏捷开发不只是管理层次的敏捷、项目参与人员的敏捷, 架构的设计也应该是敏捷的, 编程思想也应该是敏捷的, 这才是实至名归的敏捷开发。在项目真正进行过程中, 为了更好地发挥敏捷开发的优势应做到管理与技术同步规范。管理与技术是不可分割的, 二者相辅相成, 技术的敏捷使得敏捷开发方法的实施具有一定可操作性, 是敏捷开发方法的基石, 如果没有相关技术的支持, 敏捷开发是不能完全发挥效应的甚至是不可行的。只有做到管理与技术同步敏捷, 在管理的过程中得到相应技术的支撑, 敏捷开发方法才能发挥真正的敏捷, 才是真正的敏捷思想。

2.1 简单三层

敏捷开发方法要求在敏捷开发实施过程中, 如何保证开发团队的每个开发人员能够独立地进行系统模块开发是敏捷开发能否成功的关键因素。而以往的传统开发技术, 并没有将系统模块化, 整个系统的耦合性较高, 依赖性较强, 使用传统的开发技术并不能实施敏捷开发方法, 使得敏捷开发提倡的系统各模块之间并行开发成为空想。简单三层却为这一要求提供了可行性。

三层架构由底层至上层分为数据访问层 (Data Access Layer, DAL), 业务逻辑层 (Business Logic Layer, BLL), 用户界面 (User Interface, UI)^[5]。简单三层将界面的呈现、业务逻辑的处理以及访问数据库很好的分离开来, 因此在系统实现编码过程中不仅可以实现系统各模块的并行开发, 而且不要求全能型的开发人员, 只需精通其中一层即可参与到项目当中, 在保证工作高效且代码质量的同时节约了开发成本^[6]。在测试方面, 三层也可同步进行, 可以解决敏捷开发在测试环节中给项目带来的成本偏高问题, 在支持敏捷开发实施的同时, 又保证了整个系统的安

全,降低了系统的复杂性,无形之中也提高了项目参与人员的积极性,使得敏捷开发能够顺利进行。

2.2 抽象工厂模式

敏捷开发提倡的是拥抱变化,要求在开发过程中进行多次的迭代,项目团队进行周期性的交流沟通,随时应对客户的需求变化,勇敢地面对变化,对于用户的反馈,程序员要有勇气对已经编写好的代码进行适当的修改^[7]。敏捷开发里所说的勇于接受变化并不是简单的要求项目团队在客户提出新的要求时,就将之前的系统全部放弃,从头再来,这并不是真正意义上的敏捷开发。但是面对这样的需求敏捷开发只是单纯的提出了要求,并没有对此有进一步的说明和指导,而为了满足更换数据库的需求重新编译数据操作类就相当于项目从头开始,又不是最好的解决办法。如何在面对客户需求的时候,尽可能地减少代码的修改量,代码的复用性是一个关键因素。想要提高代码的复用性,就想到了设计模式,在当前的设计模式中,抽象工厂模式很好地解决了这一问题。

作为创建型模式的抽象工厂模式是 23 种设计模式中的一种,所谓创建型模式就是不需要自己实例化对象,而是由创建型模式来代替新操作^[8]。抽象工厂模式指的是提供一个创建一系列相关或者相互依赖对象的接口,而不需要指定它具体的类^[9]。该类设计模式是专门针对需求的变化来达到提高代码复用性目标的一种模式,它就相当于一个实实在在的工厂,只不过与我们现实生活中的工厂不同的是现实生活中的工厂是用来生产产品,但是这里的工厂是用来管理变化的。

使用抽象工厂模式将可变的进行封装,以接口的形式呈现,在三层中应用抽象工厂模式,在不需要修改以前代码的前提下轻松地解决了更换数据库类型这一需求,并且在系统开发完成之后甚至是使用过程中,都可轻松地更换数据库类型,有效地解决了由于需求变化导致开发周期延长,并且为系统的后期维护降低了成本。

2.3 ASP.Net MVC

客户的需求贯穿于整个项目中,虽然敏捷开发中采取先测试再编码的方式有效地应对了客户需求的变化,但是并不排除客户对测试满意,开发完成之后又提出了新要求的可能性存在。对于大多数客户来说需求的变化主要体现在系统功能和界面展示方面,针对这一可能性传统的敏捷开发并没有很好的解决办法,只能重新编写,重新测试,但是这一问题是周而复始的,这样的做法治标不治本。应对这一状况就应考虑到从技术方面入手。

在通常情况下,系统的页面开发都是用网页表格 (Web Form) 进行,虽然网页表格 (Web Form) 操作简单,可以直接拖控件对页面完成布局工作,但是它的页面展示与后台逻辑代码的耦合度较高,在遇到客户对页面布局要求变更的时候,不仅要进行页面布局的修改,还要将相应的页面逻辑进行修改,这无疑多做了很多没有必要的工作,这时 ASP.Net MVC 框架成了解决这一问题的第一选择。

ASP.Net MVC 是微软在 2009 年对外公开发布的第一

个开源的表现层框架^[10]。ASP.Net MVC 模式是一种表现模式,它可以将表现层分成模型 (Model)、控制器 (Controller) 和视图 (View) 三个组件,有效地分离了页面展示与用户界面 (UI) 逻辑代码,所以 ASP.Net MVC 是一个更加倾向于用户界面层 (UI) 的表现层框架,是网页表格 (Web Form) 的另一种选择^[11]。在面对客户对页面展示需求变化的时候,只需更改相应页面的展示效果即可。

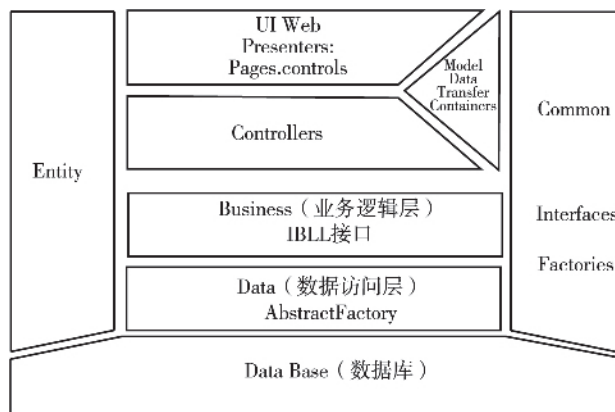
3 敏捷技术应用实例

3.1 项目简介

本校的博士研究生招生工作在 2013 年之前均是通过工作人员手工的录入以及核对完成的,因此,本校博士研究生工作存在效率低下,管理无序,数据安全性较差等诸多问题,实施办公自动化对于我校的博士研究生工作具有重大意义。本项目将采用本文所提及的融合敏捷技术的敏捷开发方法对系统进行开发。

3.2 敏捷技术在开发中的具体应用与实现

根据上述分析,博士研究生招生系统的整体架构见下图。



系统整体架构

从上图我们可以看到,本系统结合了简单三层架构、抽象工厂设计模式和 ASP.Net MVC 框架三大技术,本系统使用 2011 年发布的 ASP.Net MVC3.0 版本^[12]。本论文主要从这三个技术的应用上进行详细论述,来论证敏捷技术对敏捷开发的重要性。

3.2.1 抽象工厂模式在数据访问层的应用

对于本系统的开发主要是针对学生各种信息的管理与操作,并且录取的学生信息数据库要与我校现有的各种学生工作系统进行衔接,而所用数据库并不相同,对现在已经使用的系统进行修改并不是一个好办法,只有针对目前着手开发的博士研究生招生系统进行完善,来迎合不同数据库的需求。解决这一需求的办法就是抽象工厂模式。本文以考生登录功能为例进行技术应用说明。

创建抽象工厂类,利用 .Net 的反射机制获取数据访问层的程序集和命名空间的名称,通过数据访问接口层来

创建 user_infor 数据表的实体类, 并采用缓存技术来提高系统性能, 设置当前应用程序指定 CacheKey 的 Cache 值的核心代码如下:

```
System.Web.Caching.Cache objCache =
HttpRuntime.Cache;
```

由于数据访问层融合了抽象工厂设计模式的思想, 所以业务逻辑层调用数据访问层是通过数据访问接口层创建相对应的数据工厂实例, 并没有指定具体的数据操作类, 因此, 在面对更改数据库类型时, 只需修改配置文件的程序集和命名空间的 Value 值即可, 配置文件代码如下:

```
< add key = "DalAssemblyName" value =
"MvcBSZS.SqlDAL" />
< add key = "NameSpace" value =
"MvcBSZS.SqlDAL" />
```

由此可见, 应用了抽象工厂模式和反射机制加上配置文件的使用, 在不需要修改系统代码的前提下轻松实现了异库移植操作, 并且为本项目实施敏捷开发时适应了需求且缩短了开发周期。

3.2.2 ASP.Net MVC 框架的应用

在 ASP. Net MVC 中 UI 逻辑在 Controller 组件中进行编译, Controller 负责将数据从 Model 取出传递给 View^[13]。在本系统里, ASP. Net MVC 代替网页表格 (Web Form) 作为三层中的用户界面层 (UI), 所以在界面展示编译中, ASP. Net MVC 框架中的控制器 (Controller) 组件则负责与业务逻辑层进行对话, 从业务逻辑层调用校验用户的方法来获取数据库中的考生登陆信息。判断登陆是否成功的部分代码如下:

```
IBLL.Iuser_inforService user_inforService = new user_
inforBLL ();
public ActionResult ProcessLogin ()
{
string loginid = Request [ "LoginCode" ];
string password = Request [ "LoginPwd" ];
user_infor model = user_inforService. Getuserinforby-
loginid ( loginid, password );
if ( model [FY = ] null )
{
return Content ( "用户名不存在, 请确认是否注册!
若没有注册请先进行注册" );
}
if ( model.pwd != CommonHelper. GetMd5stringFromString ( password ) )
{
return Content ( "密码错误" );
}
Session [ "LoginUser" ] = model;
return Content ( "ok" ); //登陆成功, 前台调用 Ajax
进行页面跳转
}
```

前台的页面布局交由视图 (View) 组件负责, 由于

它与控制器 (Controller) 组件的低耦合性, 视图 (View) 的页面展示只需单纯的 Html 标签即可实现, 并且不需要将标签 ID 传到后台, 实现了 UI 展示与 UI 逻辑的彻底分离。登录页面的视图 (View) 核心代码如下所示:

```
< td align = right class = "style17" > 身份证号: < /
td >
< input type = "text" name = "LoginCode" style =
"width: 111px" />
< td align = right class = "style13" >
< div id = loginText > 密码: < /div > < /td >
< td class = "style8" > < input type = "password"
name = "LoginPwd" style = "width: 139px" /> < /td >
< td align = right class = "style8" >
< div > &nbsp;
< input type = "submit" id = "go" value = "登录" /
> < /div > < /td >
```

从以上代码我们可以看出, 视图 (View) 页面都是由 Html 标签实现, 所以在更换页面布局的时候就不需要考虑前台标签与后台逻辑的绑定问题, 不需要修改用户界面 (UI) 逻辑代码。

4 结 论

敏捷开发是一种以需求为导向的驱动开发方法, 需求的变化会给整个系统带来副作用, 所以需要自动化去支持变化, 在应对变化的同时保证敏捷开发方法在项目实施的成功率和系统质量, 缩短开发周期, 解决敏捷开发在管理上不能解决的问题。结合了简单三层、抽象工厂模式和 ASP. Net MVC 开源框架的敏捷开发使得开发团队在多变的环境下按时按质按量地完成了本校博士研究生招生系统的开发。由此可见, 结合了敏捷技术的敏捷开发将敏捷开发方法发挥到极致甚至在敏捷开发方法的基础上进一步缩短项目周期, 提高软件质量和软件的维护性、扩展性。

参考文献:

- [1] James Shore, Shane Warden. The Art of Agile Development [M]. Sebastopol: O'Reilly Media, 2009.
- [2] High Smith J. Agile Software Development Ecosystems [M]. Beijing: China Machine Press, 2004.
- [3] 王世安. 基于敏捷开发的构件化软件可靠性研究 [J]. 微电子学与计算机, 2011 (3): 119 - 122.
- [4] 陈楠, 陈文培. 敏捷开发中相关技术的应用 [J]. 计算机应用与软件, 2011 (4): 202 - 204.
- [5] C. Thilmany. . NET Patterns: Architecture, Design, and Process [M]. Boston: Addison - Wesley Professional, 2003.
- [6] Eric J. Braude. NEW Software Design: From Programming to Architecture [M]. Beijing: Publishing House of Electronics Industry, 2005: 23 - 26.
- [7] Alistair Cockburn. Agile Software Development [M]. Boston: Addison Wesley, 2001: 30 - 45.
- [8] Markus Aleksey, Axel Korthaus. Supplementing the Well - Known Factory Pattern for Distributed Object Oriented Systems [C]. Symposium on Distributed Computing and Applications for Business Engi-

neering and Science, 2002 (12): 105-108.

[9] 武光明. 设计模式在全球化软件开发中的应用 [J]. 计算机应用与软件, 2014 (1): 9-17.

[10] Sanderson S. Pro ASP. NET MVC Framework [M]. Berkeley: Apress, 2009.

[11] Connery R, Hanselman S, Hack P, et al. Professional ASP. NET MVC 1.0 [M]. Hoboken: Wiley Publishing, 2009.

[12] Steven Sanderson, Adam Freeman. Pro ASP. NET MVC 3

Framework [M]. Berkeley: Apress, 2011.

[13] 周文红. 基于 ASP. NET MVC 框架的 Web 应用开发 [J]. 计算机与现代化, 2013 (10): 197-199.

~~~~~  
[作者简介] 戴洪蕾, 硕士研究生, 主持内蒙古自治区科研创新项目“基于工厂模式的敏捷开发的辨析与应用研究”(S20141012814); 王建国, 副教授。研究方向: 信息管理与信息系统、项目管理。

~~~~~  
(上接 P49) 利用 Eviews3.0 软件分析则会得到残差 e_t 与 $e_{t-1}, e_{t-2}, \dots, e_{t-10}$ 的各期相关系数和偏相关系数, 其中 Autocorrelation 和 Partial Correlation 两个指标所对应的部分在第一期和第二期并没有超过虚线的部分, 别的指标也都处于正常范围内。由此我们可以知道该回归模型不存在自相关。

所以该模型的最终形式为:

$$Y = -113844.6 + 11310.94 \ln X_1 + 0.373973 X_2$$

$$(9470.905) (853.9566) (0.002122)$$

$$t = (-12.02046) (13.24533) (176.2051)$$

$$R^2 = 0.999983 \quad \bar{R}^2 = 0.999981$$

$$F = 585374.1 \quad n = 24$$

6 模型的总结与建议

由以上结果可以得出在经济发展中, 对我国的人民币储蓄存款的主要影响因素是国内生产总值的提高, 即国家经济的快速发展, 促使了储蓄的不断增长。货币供应量只是货币政策的一种调节方法, 只是对经济起到辅助作用, 并不能从根本上和实质上影响居民储蓄情况。所以, 居民生活水平的提高, 即居民储蓄的增加更多的要依靠国民经济整体的发展和完善。但在使用时, 要结合模型和人民币

储蓄存款这一经济变量本身的经济特性, 因为金融和经济形势复杂, 且全球经济一体化不断加剧, 更加增加了我国经济的复杂性。除此之外, 模型是动态的, 在不同时期内, 影响它的主要因素不一定相同。如有可能的话, 此模型还能应用于对于某些省市的居民储蓄存款的预测。

参考文献:

[1] 中国国家统计局. 中国统计年鉴 [M]. 北京: 中国统计出版社, 2014.

[2] 张晓峒. 使用指南与案例 [M]. 北京: 机械工业出版社, 2007.

[3] 孙敬水. 计量经济学 [M]. 北京: 清华大学出版社, 2003.

[4] 庞皓. 计量经济学 [M]. 北京: 科学出版社, 2007.

[5] 易丹辉. 数据分析与 Eviews 应用 [M]. 北京: 中国统计出版社, 2002.

[6] 李子奈. 计量经济学 [M]. 3 版, 北京: 高等教育出版社, 2010.

[7] 姜启源等. 数学模型 [M]. 3 版, 北京: 高等教育出版社, 2003.

[8] 曾五一. 统计学简明教程 [M]. 1 版, 北京: 中国人民大学出版社, 2012.