

# 基于敏捷思想的 iOS 平台软件的设计与实现

段学东<sup>1</sup>, 何九周<sup>2</sup>

(1. 武汉理工大学华夏学院, 湖北 武汉 430223;

2. 武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)

**摘要:**软件的开发过程与最初的计划保持一致已经不再是当今软件开发的主要目标了, 如何应对客户对于需求的不断变化, 快速地交付高质量的软件并做到让客户满意, 已经成为了一个软件开发公司能否在当今竞争激烈的软件行业中得以生存的关键。文中在对统一软件过程以及敏捷方法的优势和局限性研究与分析的基础上, 利用敏捷思想对统一过程进行适当裁剪并利用统一思想对敏捷开发进行合理扩充, 提出了一种基于统一过程和敏捷方法的软件开发过程, 成功地将敏捷过程应用到 iOS 智能手机操作系统开发这一实际的项目中。实践证明, 敏捷开发方法在应对需求频繁变化和软件开发效率方面均起到了很好的效果。

**关键词:**敏捷方法; iOS; XP; Scrum

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1673-629X(2012)09-0053-06

## iOS Platform Software Design and Implementation Based on Agile Development Methods

DUAN Xue-dong<sup>1</sup>, HE Jiu-zhou<sup>2</sup>

(1. Huaxia College of Wuhan Univ. of Tech., Wuhan 430223, China;

2. College of Computer Science and Technology, Wuhan Univ. of Tech., Wuhan 430070, China)

**Abstract:** Making the process of software development keep consistent with the primal plan is no long the primary goal of nowadays software development. How to response the continuous requirement changes from the customers, frequently deliver software with high quality and make the customers satisfied already become the key factor which a software corporation survive in the jungle of software industry and the research focus in the field of software engineering. Based on the research and analysis of advantages and limitations of rational unified process and agile development methods, making use of the idea of agile development methods to cut out the rational unified process properly and making use of the thinking of rational unified process to expand agile development methods logically, successfully applying agile development methods to iOS smart phone operating system that is an actual project. Practice proves that agile development methods has made perfect result both in response to frequent changes in demand and increasing the efficiency of software development.

**Key words:** agile method; iOS; XP; Scrum

## 0 引言

随着科技发展的日新月异, 3G 网络的建成运行和 3G 智能手机的大众化, 智能移动设备用户对无线互联网应用的需求将越来越大。基于 iOS 平台的 iPhone 手机作为当今市场用户体验最好的智能移动设备, 也对互联网应用提出了越来越多的需求。

iOS App Store 开创了一个全新的软件生态系统, 它不仅改写了软件的交付和消费方式, 也对软件的设计产生着显著的影响。

App Store 上成功的应用, 要求的设计过程要比传统的(以用户为中心的设计)UCD 过程更敏捷; 截至 2012 年 1 月 App Store 提供 500,000 多款应用软件。成功的 App 设计, 要求在上线第一天就能够吸引用户。如果你上市的第一个月没有进入排行榜, 那第二个月就会消失的无影无踪。

文中结合了“iOS 平台的新闻发布和阅读系统”的项目, 设计并实现了基于敏捷设计流程的应用系统。

## 1 敏捷开发概述

敏捷开发起源于 20 世纪 30 年代的一些项目如美国航天局水星计划, 最早有记载的使用迭代和增量开发的主要项目之一, 是在 20 世纪 70 年代为第一艘美国三叉戟潜艇开发的第一指挥和控制系统。该项目有

收稿日期: 2012-01-17; 修回日期: 2012-04-23

基金项目: 武汉理工大学重点计划项目(104-611002381)

作者简介: 段学东(1972-), 男, 硕士, 工程师, 研究方向为数据库与软件工程。

大约一百万行代码,进行得非常成功。1976 年 Tom Gilb 在他的著作《软件度量》(“Software Metrics”)一书中阐述了他的迭代和增量开发实践。20 世纪 90 年代推荐使用迭代和增量开发方法的资料和文献显著增加。2001 年二月一组由 17 位在 DSDM, XP, Scrum, FSD 等领域的专家组成的代表团齐聚美国犹他州,寻找这些方法的共同点。最终,这些专家制定并宣布了敏捷开发宣言,由此形成了现在人们所认识的敏捷开发和后来的敏捷联盟<sup>[1]</sup>。

敏捷开发是一种以人为核心,迭代、循序渐进的开发方法。在敏捷开发中,软件项目的构建被切分成多个子项目,各个子项目的成果都经过测试,具备集成和可运行的特征。简言之,就是把一个大项目分为多个相互联系,但也可独立运行的小项目,并分别完成,在此过程中软件一直处于可使用状态。

敏捷开发由几种轻量级的软件开发方法组成。它们包括:极限编程(XP),Scrum,精益开发(Lean Development),动态系统开发方法(DSDM),特征驱动开发(Feature Driver Development),水晶开发(Cristal Clear)等等<sup>[2]</sup>。敏捷开发方法可以降低需求变化的成本。其定义了诸如编写用户案例,架构规范,实施规划,迭代计划,代码开发,单元测试,验收测试等等一整套简单的开发流程。敏捷开发方法核心做法为小规模,频繁的版本发布,短迭代周期,提倡开发成员之间互动交流、反馈、简单、勇气和团队。

敏捷开发原则和方法包括:迭代式开发,即整个开发过程被分为几个迭代周期,每个迭代周期是一个定长或不定长的时间块,每个迭代周期持续的时间一般较短,通常为一到六周;增量交付,产品是在每个迭代周期结束时被逐步交付使用,而不是在整个开发过程结束的时候一次性交付使用,每次交付的都是可以被部署到用户应用环境中被用户使用的、能给用户带来即时效益和价值的产品;开发团队和用户反馈推动产品开发,敏捷开发方法主张用户能够全程参与到整个开发过程中,这使需求变化和用户反馈能被动态管理并及时集成到产品中,同时,团队对于用户的需求也能及时提供反馈意见,新的功能或需求变化总是尽可能频繁地被整合到产品中,一些项目是在每个迭代周期结束的时候集成,有些项目则每天都在这么做;要求拥有一个积极的、自我管理的、具备自由交流风格的开发团队,是每个敏捷项目必不可少的条件,人是敏捷开发的核心,敏捷开发总是以人为中心建立开发的过程和机制,而非把过程和机制强加给人<sup>[3]</sup>。

敏捷软件开发是一种用于开发软件的新型管理模式,用来替代传统的以文档驱动开发的瀑布开发模式。敏捷软件开发价值观包括:个人的能力和相互的交流

胜过过程和工具;设计出可以正常工作的软件胜过面面俱到的文档;与客户相互合作胜过合同谈判;对事件的快速响应变化胜过循规蹈矩。敏捷开发方法的核心思想概括起来就是:“适应变化”和“以人为本”。

## 2 iOS 操作平台

iOS 又称 OS X iPhone,是由苹果公司为 iPhone 开发的操作系统。后来陆续套用到 iPod touch、iPad 以及 Apple TV 产品上。iPhone 最初使用的是特别定制的 Mac OS X 操作系统,发售初期自带系统版本为 1.0.2,之后以固件更新的方式修正低版本的错误并且引入新的功能。目前最新的版本是 2011 年 10 月 12 日发布的 iOS 5.0 最终版(公开版)。iOS 操作系统占用大概 512MB 的内存空间。

iPhone 和 iPod Touch 使用基于 ARM 架构的 CPU,因此,所有其他操作系统下的应用程序不能直接在 iOS 上运行。需要针对 iOS 的 ARM 重新编写并编译后,才能在 iPhone 下运行<sup>[4]</sup>。

iOS 为封闭的操作系统,苹果公司对外不公布其操作系统源码。

iPhone 手机未配置键盘,所有对手机的操作均通过手机的触摸屏来操作。iPhone 手机采用先进的电容式触摸屏,支持多点触控的方式进行操作。iPhone 与系统交互的操作包括:滑动,轻按,挤压,旋转,并且可以支持其旋转装置令屏幕改变方向。

启动 iPhone 应用程序的唯一方法就是在菜单目录中轻点程序的图标,程序运行后可屏幕触摸、点击对程序进行控制。退出运行中的程序则是按下屏幕下方的 home 键。在第三方软件退出后,它直接就被关闭了。

iOS 是以 Darwin 为基础的。iOS 的系统架构分为四个层次:核心操作系统层(the Core OS layer),核心服务层(the Core Services layer),媒体层(the Media layer),可轻触层(the Cocoa Touch layer)。iOS 体系结构与 Mac OS X 的基础架构类似。开发者创建的应用程序不会直接与硬件进行交互,而是与相对应的驱动器进行连接。这样的做法可以消除开发者应用程序在不经意间改变了底层的硬件设备的危险。

iPhone 是为移动终端而开发,要解决的用户需求与 Mac OS X 不同。在底层的实现上 iPhone 与 Mac OS X 共享了一些底层技术,iOS 使用软件栈,在堆栈的底层是 Mach 内核和一些硬件驱动器,管理程序的执行;在底层之上的其他层包括开发者必须的核心技术和接口;iOS 在内核和驱动级没有任何接口,在软件栈的更上层为开发者设置了许多技术接口。

苹果公司将大部分的系统接口都收集在一个叫做

框架 (framework) 的程序包中。一个 framework 包含了一个动态共享库以及支持该库的一些资源 (如头文件、图片、帮助程序等)。使用 framework, 应用程序开发人员可以将它内联到开发工程内, 从而使其得以访问 framework 的特性, 并指导开发工具去哪里查找头文件和 framework 的其他资源。除了 framework, Apple 还在它的标准共享库 (Standard Shared Libraries) 中提供其他技术。iOS 是基于 UNIX 的, 构成操作系统的底层架构的技术都属于开源代码技术。这些技术的接口就储存在标准库以及接口目录中。

由于 iOS 是从 Mac OS X 核心演变而来, 因此开发工具也是基于 Xcode。该 SDK 可分类为触控、媒体、核心服务、OS X 核心 4 大项。和 Xcode 工具一样, SDK 开发工具中包含的 iPhone 模拟器用来在开发人员的电脑上模拟 iPhone 的外观和感觉。该 SDK 需要拥有英特尔处理器且运行 Mac OS X Leopard 系统的 Mac 才能使用。

### 3 基于敏捷思想的 iOS 平台软件设计流程和基本原则

基于敏捷思想的 iOS 平台软件的设计流程, 跟设计传统 web 软件的流程没有太大差别, 都是基于统一的设计方法论。App Store 上成功的应用, 绝大部分都是面向个人的软件, 它们功能相对简单, 注重满足用户的核心需求, 设计上极力追求完美<sup>[5]</sup>。

基于敏捷思想的 iOS 平台软件的设计流程, 可以归纳为以下步骤: 市场定位, App 定义 (ADS), 概念草图, 概要原型, 详细原型, 设计说明和整理, 设计开发。

在这个流程中, 并没有传统 UCD 方法论中强调的用户分析、场景分析、信息架构设计等环节, 它们已经变成基本原则, 融入到具体的原型设计过程中去了。因为软件足够小, 不需要也不可能承受冗长的基础分析和设计过程所带来的成本, 它需要的是更敏捷的设计流程, 用尽量完美的设计, 来满足用户的特定需求<sup>[6]</sup>。

同样的, 敏捷设计流程, 逼迫设计团队必须裁剪需求, 才能更好地适应赢家通吃法则。一个小软件的失败, 损失的可能只是 4 周的工作时间, 这并没有什么大不了, 可以通过另一个新产品来获得成功<sup>[7]</sup>。

## 4 iOS 平台的新闻发布和阅读系统的设计与实现

### 4.1 市场定位

首先, 要确定你要做的程序还没有人做过。如果你发现已经有类似的应用程序, 那你需要比它做的更

好, 有一些独特的优化设计。最好的调查方式是到 iTunes Store 上搜索已有的 iPad 程序。当有了创意, 你还需要有个明确的定位, 它会在后续的设计过程中决定应用程序的设计要点。应用程序的定位可以通过苹果的人机界面指南 (Human Interface Guidelines) 图来确定<sup>[8]</sup>。

在《iOS 平台的新闻发布和阅读系统》这个项目中, 使用简单的辅助工具, 特点明显, 精美易用的界面设计, 让这个项目与其他竞争者明显区分开来。

### 4.2 App 定义 (ADS)

iPhone 平台有三种类型的应用: 效率型应用 (Productivity Applications)、实用工具 (Utility Applications)、沉浸式应用 (Immersive Applications)。每一种都有各自不同的特点和应用场合。设计之前如果能够清楚产品的目标和特点, 选择合适的应用类型可以更好地应对。效率型应用用于帮助完成日常的一些工作, 如收发邮件、即时通信、照片管理与分享。用户快速查看、跳转、执行、完成, 连贯的动作要求界面简单直接, 让用户可以全神贯注于任务本身。这类应用一般都附带自然的层级结构。所以, 表格视图 (table view) 可以在这类软件里大派用场。实用工具用于执行一项简单的任务 (如计算器, 天气报告), 简单, 容易配置就可以了。沉浸式应用可以为用户带来极致的娱乐和游戏体验, 这类应用可以进一步细分为游戏和虚拟仿真类。这类应用和标准的系统界面不同, 用户希望这类应用能够给他们带来最大的娱乐, 因此, 界面设计的自由发挥度比较高, 仿真的、可爱的设计风格往往能恰到好处。

确定应用程序定位后, 接下来需要聚焦这个应用程序的核心功能。在团队合作设计时, 这一点尤其重要。团队在提出各种功能需求时, 很容易陷入哪些功能要包含在第一个版本中的争论。Apple 把这个过程叫设计 ADS (Application Definition Statement), 或者叫设计精简的 ADS。

### 4.3 概念草图

前面的过程, 仅仅是设计的导入阶段。接下来要做的, 是产品草图设计。按照设计构想, 勾画出用户要用到的界面, 包括像按钮之类的界面交互元素; 筛选出核心用户最常用的、最适合移动应用场景的功能<sup>[9]</sup>。

要设计的工具软件, 通常, 它只需要主界面和一个在背面显示相关信息的辅助界面, 它通过信息按钮触发后翻转显示。如果设计的是其他 App, 可能还需要更多的界面。重点是要设计界面与界面之间的切换方式, 这一点在设计交付给开发人员时会显得尤其重要。这个过程叫做 App 功能穿越 (App Functionality Walk-through)。



#### 4.4 概要原型设计

概要原型设计的重点是不要陷入过多的细节,概要原型只是把设计的概念草图数字化,便于在电脑上持续的改进。所以,尽量使用黑白、粗糙的线条和图形来制作概要原型,别在细节上纠结,浪费时间。

#### 4.5 详细原型设计

概要原型设计完成后,开始设计注重细节和精度的详细原型设计。使用 PhotoShop,也可以选用自己熟悉的其他工具。一般,为 iPad 设置尺寸为 1024×768 的画布,然后根据概要原型进行细节设计。

当设计《iOS 平台的新闻发布和阅读系统》应用程序的主界面时,用来显示新闻阅读器的两大功能,新闻在线、偏好设置是设计的重点,它不需要包含所有功能。新闻在线用于切换到新闻频道列表界面;鉴于 Apple 提倡有质感、有仿真度的图形界面,偏好设置用于设置该新闻阅读器新闻显示的样式。很重要的一点是,应用程序所展现的信息,必须简洁明了,没有多余的文字。所以,在界面设计上,没有必要引入任何华丽的图形或者其他的信息来干扰用户,让用户能一眼就看出应用程序的用途。在数据条目之间使用间隔色,用醒目的字体内容,这些都是很好的设计体现。主界面背后的相关信息界面,使用 Apple 的标准界面即可,为用户提供搜索后加入关注的功能。重要的是,确保所有的信息都一目了然,不隐晦,不误导用户。

应用程序图标(icon)是指放在 iPhone 屏幕上用来启动程序的图标。iPhone 的屏幕大小只有 480 \* 320px,在这狭小的屏幕里,用户希望放下尽可能多的图标,这些图标必须具有突出的视觉效果才能方便用户辨认。icon 的设计将决定应用程序在 App Store 上的辨识度。可以从简单的轮廓设计开始,先把核心创意表现出来。除非有必要,icon 最好不要包含文字,尽量使用跟应用程序图形界面一致的材质和渐变。

#### 4.6 设计资料整理

把界面和描述集中到一张大图,尽可能地把所有可遇见的情况都描述清楚。将概要原型设计,所有的图形界面设计图(一般是 PSD)和图标打包在一起,做上清楚的标注,供开发阶段人员使用。

#### 4.7 设计开发

《iOS 平台的新闻发布和阅读系统》是根据无线互联网通信技术发展要求应运而生,可使手机用户获得即时资讯服务。文中以新闻订阅模块为例进行设计和实现。

系统设计开发阶段的任务就是根据系统设计说明书中规定的各种功能,考虑实际条件,具体设计实现逻辑模型的技术方案,设计新系统的物理模型。结合系统规划和系统分析阶段的功能列表,利用 UML 工具制

作了该系统的用例图<sup>[10]</sup>。

在系统用例图中(见图 1)介绍了系统的全部功能,描述了用户对新闻订阅服务可以执行的功能,例如增加、修改等工作。

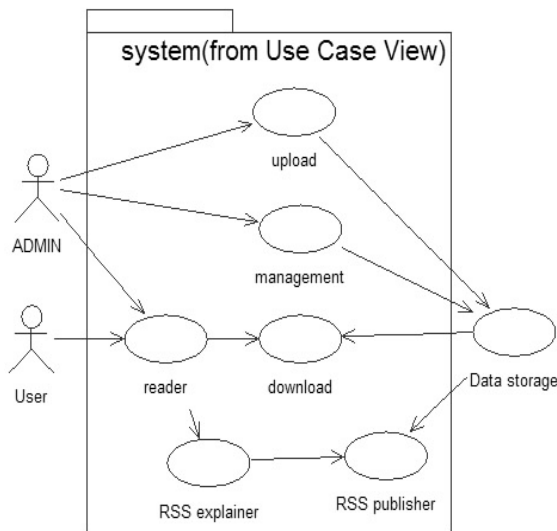


图 1 系统用例图

通过用例图等资料,可以根据其进行交互图的绘制。顺序图是交互图中最常用的一种,交互图是用例的具体实现。其实现是以对象和对象间的协作为基础的,因此通过交互图可以严格地对照检查系统的每一项功能需求是否都得到满足,这些需求都落实到哪些对象,以及这些对象是如何来解决的。整个过程是连贯的,从用例模型到分析模型到设计模型之间有一致性和可追踪性。

顺序图的描述涉及用例实现的多个对象实例以及对象交互时相互传递的消息。并按照用例的执行步骤为顺序指明对象的交互顺序。

图 2 是用户界面中对新闻来源进行管理的界面顺序图,用户界面被设计成多层结构。

新闻管理列表模块设计的主要功能:将相关主题中所有频道罗列出来,可以对其进行添加、删除等操作。通过点击选中频道,可进入此频道的用户设置界面。本模块中用到的,所有的频道数据都保存在频道列表文件中,此模块界面中显示的数据,也全部来自频道列表文件,需要对此文件进行读写等操作。按照用户选择的相关选项,将选中频道的数据从文件中读出,显示到用户界面。主要实现代码:

频道界面列表显示代码:

```
cell.textLabel.text = [[channellist allKeys] objectAtIndex:indexPath.row];
```

```
cell.accessoryType = UITableViewCellAccessoryDetailDisclosureButton;
```

频道选择界面显示代码:

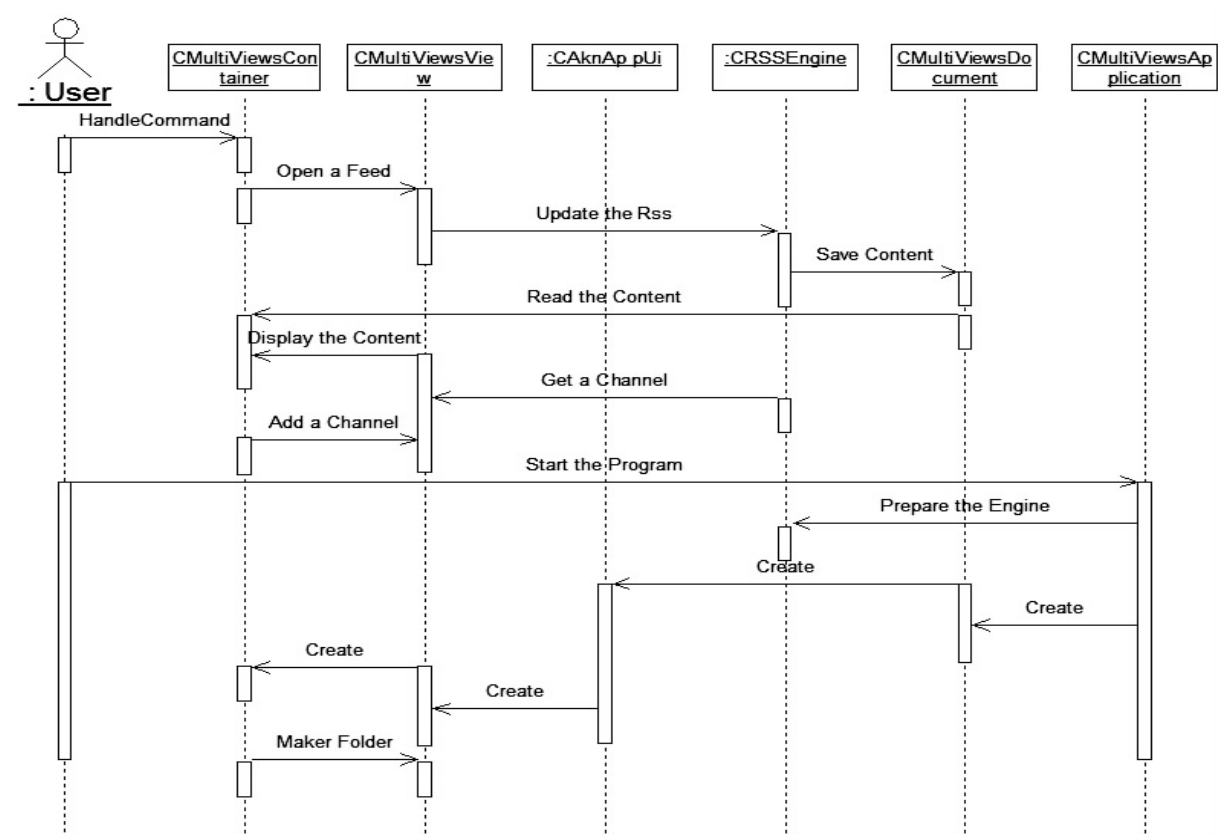


图2 新闻管理界面顺序图

```
-(void)tableView:(UITableView *)tableView didSelectRowAtIndex:(NSIndexPath *)indexPath {
    NewsListViewController * anotherViewController = [[ NewsListViewController alloc ] initWithNibName:@" NewsListViewController" bundle:nil];
    anotherViewController.istyle = [[ systemdata objectForKey:@" style"] intValue];
    anotherViewController.iurl = [ channellist objectForKey:[ tableView cellForRowAtIndexPath:indexPath].textLabel.text];
    anotherViewController.title = [ tableView cellForRowAtIndexPath:indexPath].textLabel.text;
    [ self.navigationController pushViewController: anotherViewController animated:YES];
    [ anotherViewController release];
}

频道添加的代码:
-(void)showaddviewController{
    AddMyChannelViewController * addviewController = [[ AddMyChannelViewController alloc ] initWithNibName:@" AddMyChannelViewController" bundle:nil];
    NSString * atitle = [ [ NSString alloc ] initWithFormat:@"% %@主频道",[ systemdata objectForKey:@" mainchannel" ]];
    addviewController.title = atitle;
    addviewController.channelkey = [ systemdata objectForKey:@" mainchannel" ];
    [ atitle release];
    addviewController.supresource = channellist;
```

```
[ self.navigationController pushViewController: addviewController animated:YES];
[ addviewController release];
}
```

《iOS 平台的新闻发布和阅读系统》程序可以实现对选择新闻的发布和阅读,并且可以实时的以手机浏览器的形式连接新闻网址,查看网站提供图片、音频以及相关视频等功能。

5 结束语

传统的软件开发模式将项目的生命周期明确地划分为几个阶段,完成一个阶段才进入下一个阶段。在项目初期希望细化所有的需求,并希望在一个阶段将需求固定后不再改变。在需求定义完毕后,在编码之前进行较详细的预先设计,完成所有或者大部分的设计工作才开始编码。每一个阶段需要产出大量的文档作为下一阶段的输入。由于现代软件系统的功能和设计越来越复杂,市场需求变化较快,所以传统的软件开发模式所要求的严格地完成一个阶段再进入下一个阶段太过理想化。针对传统开发模式所存在的问题,业界提出的解决方法就是应用迭代开发方式,而敏捷开发更是在迭代开发的基础上,作了进一步的改进,敏捷开发方法由于它应用迭代和增量的开发模式,所以可以看作是经过改进的迭代开发方法<sup>[11]</sup>。

敏捷开发注重实际可用的软件,文档简单,对于变

化随机应变,对于每一步都有十分清晰的计划,从需求调研开始,整个开发团队就都进入了忙碌的阶段;需求、设计、代码、测试。而瀑布模型则是一步一步的走下去的。敏捷开发注重“民主”,每个开发中作者都是自己的领导者。近年来,随着敏捷开发思想的提出,以及 UP(Unified Process,统一流程)、敏捷 UP、Scrum 和 XP(极限编程实践)等一系列的实践方法得到应用,迭代、增量的开发模式得到了更多的赞誉声音,目前,最为热门的是以 Scrum 和 XP 进行组合的敏捷开发方式,已经被腾讯、华为、上海贝尔等一些大公司所采用。

敏捷开发方法是一个过程,是一个持续的应用原则、模式以及实践来改进软件的结构和可读性的过程。它致力于保持系统设计在任何时间都尽可能得简单、干净和富有表现力<sup>[12]</sup>。

并不是选用某种开发模式,或者更前进的开发技术,就一定能保证将项目做成功。而是参考业界的一些成功的最佳实践活动经验,不断地对所用的开发模式进行检讨,合理地调整开发过程,在现有的开发资源的情况下不断地将项目的开发过程发挥得更高效。

参考文献:

[1] Livermore J A. Factors that impact implementing an agile software development methodology [ C ]//Proceedings of 2007

(上接第 52 页)

知识是进行决策规划的关键。知识获取是构造专家系统的“瓶颈”问题,专家知识的好坏直接影响整个系统的性能,因此知识获取方法得到了广泛的研究和应用。属性约简是知识获取的关键步骤,可以从一定程度上消除决策系统中的条件属性<sup>[13,14]</sup>。文中运用粗糙集理论,通过粗糙集的约简消除了冗余的条件属性,实现了对知识库的精简。进行属性值的约简相对简单,属性约简则困难很多。结果表明,该方法可以简化复杂系统的结构,并能有效维护知识库的结构和性能。

参考文献:

[1] 裴小兵. 粗糙集的知识约简研究[ D ]. 武汉: 华中科技大学, 2006.

[2] Giarratano J C, Riley G D. 专家系统原理与编程[ M ]. 北京: 机械工业出版社, 2006.

[3] 任永昌. 软件成本估算及其专家系统研究[ D ]. 阜新: 辽宁工程技术大学, 2008.

[4] 关欣, 衣晓, 何友. 一种新的粗糙集属性约简方法及其应用[ J ]. 控制与决策, 2009, 24(3): 464-467.

[5] 王彪, 段禅伦, 吴昊, 等. 粗糙集与模糊集的研究及应用[ M ]. 北京: 电子工业出版社, 2008.

[6] Dai Jianhua, Li Yuanxiang, Liu Qun. A hybrid genetic algo-

IEEE Southeast Conference. [ s. l. ]: [ s. n. ], 2007: 82-86.

[2] 赵剑冬, 林建. 敏捷方法在软件项目开发中的实践[ J ]. 计算机工程与设计, 2007(6): 2772-2774.

[3] 夏显鄂, 梁洪峻. 敏捷软件开发与计划驱动开发的概述比较[ J ]. 计算机工程与设计, 2007(8): 4035-4037.

[4] 何伟, 杨宗德, 张兵. 基于 Symbian OS 的手机开发与应用[ M ]. 北京: 人民邮电出版社, 2006.

[5] 罗运模. 软件能力成熟度模型集成(CMMI)培训教程[ M ]. 北京: 清华大学出版社, 2003: 106-130.

[6] 施瓦伯. Scrum 敏捷项目管理[ M ]. 北京: 清华大学出版社, 2007: 100-115.

[7] 马丁. 敏捷软件开发: 原则、模式与实践[ M ]. 北京: 人民邮电出版社, 2008.

[8] Starting an iPhone Application Business For Dummies[ M ]. [ s. l. ]: Wiley, 2010.

[9] Sutherland J, Schoonheim G, Rustenburg E, et al. Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams[ C ]//Agile 2008 Conference. [ s. l. ]: [ s. n. ], 2008.

[10] Moore R, Reff K, Graham J, et al. Scrum at a Fortune 500 Manufacturing Company[ J ]. AGILE, 2007(8): 175-180.

[11] 成奋华, 金敏. 基于敏捷过程的 IT 项目范围管理的研究与应用[ J ]. 计算机技术与发展, 2010, 20(10): 233-236.

[12] 沈雷, 沈建军. 敏捷方法的研究与实践[ J ]. 计算机工程, 2005(4): 219-222.

rithm for reduct of attributes in decision system based on rough set theory[ J ]. Wuhan University Journal of Natural Sciences, 2002, 7(3): 285-289.

[7] 史琨. 粗糙集的知识获取方法研究[ D ]. 太原: 山西大学, 2009.

[8] 张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法[ M ]. 北京: 科学出版社, 2003.

[9] 代广珍, 徐超. 基于 RS 理论的快速属性约简求核方法[ J ]. 计算机技术与发展, 2011, 21(4): 133-135.

[10] 颜艳. 基于粗糙集的属性约简算法及其应用研究[ D ]. 无锡: 江南大学, 2008.

[11] Hedar Abdel-Rahman, Wang Jue, Fukushima M. Tabu search for attribute reduction in rough set theory[ J ]. Soft Computing, 2008, 12(9): 909-918.

[12] Deng Tingquan, Ma Minghua, Wang Xinxia, et al. An improved approach to attribute reduction with ant colony optimization[ J ]. Fuzzy Information and Engineering, 2010, 2(2): 145-155.

[13] 沈玮, 赵佳宝. 一种新的启发式粗集决策表属性约简算法[ J ]. 计算机技术与发展, 2010, 20(10): 16-20.

[14] 汪凌, 胡培. 基于粗糙集的决策系统知识获取算法及实证分析[ J ]. 情报杂志, 2009, 28(3): 144-147.

# 基于敏捷思想的 iOS 平台软件的设计与实现

作者:	<a href="#">段学东</a> , <a href="#">何九周</a>
作者单位:	<a href="#">段学东(武汉理工大学华夏学院, 湖北 武汉 430223)</a> , <a href="#">何九周(武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)</a>
刊名:	<a href="#">计算机技术与发展</a>
英文刊名:	<a href="#">Computer Technology and Development</a>
年, 卷(期):	2012(9)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201209016.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201209016.aspx)