

敏捷开发平台的设计

李 新

(中国科学院 计算机网络信息中心 ARP 中心, 北京 100190)

摘 要: 敏捷开发平台是支持敏捷过程的软件开发平台, 敏捷开发平台通过生成 Java 代码和读取存放在数据库中的配置数据, 实现系统的可配置。这个平台包括一系列敏捷开发工具, 敏捷组件、敏捷表单、敏捷查询等; 同时, 敏捷开发平台还实现了一套敏捷 workflow 系统和敏捷数据交换平台。敏捷开发平台对数据库的设计做了一些标准规范, 有利于数据资源的有效利用。敏捷开发平台实现了系统设计标准化, 能够支持在大型信息化项目中采用敏捷开发过程。

关键词: 敏捷开发; 软件开发平台; 系统设计标准化; 工作流; 数据交换平台

中图分类号: TP311.52 **文献标识号:** A **文章编号:** 1000-7024 (2012) 09-3604-05

Design of agile development platform

LI Xin

(ARP Center, Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Agile development platform support the agile processes of software development, and enhance the system's configuration by generating java code and read configuration data stored in the database. This platform includes a series of agile development tools, agile component, agile form, agile query, etc. Agile development platform have realized a set of agile workflow system and agile data exchange platform. Agile development platform has done some standard specification for database design, facilitates the effective use of data resources. Agile development platform realize system design standardization, and support using agile development process in large information projects.

Key words: agile development; software development platform; system design standardization; workflow; data exchange platform

0 引 言

软件工程的兴起源于 20 世纪 60 年代。软件开发方法的发展大致经过以下几个阶段: 瀑布法、原型法、迭代法、敏捷法。从软件开发方法的发展趋势来看, 软件开发方法已经由“自顶向下逐步细化”发展到“反复迭代逐步逼近”^[1]。

敏捷法起源于 20 世纪 90 年代, 是一种以人为核心、反复迭代、循序渐进的开发方法。敏捷法是“适配性”而非“预设性”。即敏捷方法始终认为用户的需求是可变的, 依照需求的可能变化来设计软件^[2]。

敏捷法一出现就受到业界的欢迎, 特别适应于那些需求不断变化的信息化系统。但是, 敏捷开发需要软件开发技术的支持, 没有可灵活配置的软件开发平台, 敏捷过程的成本将难以控制。

1 敏捷开发平台的设计

随着软件在规模、复杂度、功能上的极大扩展和提高,

平台化已经成为软件开发的必然发展趋势。敏捷开发要求面临不断变化的用户需求, 快速开发出高质量的软件系统, 因此, 必须有能够支持敏捷过程的软件开发平台。

1.1 敏捷开发平台的实现原理

敏捷开发的软件要求系统可配置, 系统的配置一般是通过一些参数的设置来实现的, 当可配置参数过多时, 会非常复杂, 给管理带来困难, 而且还有可能影响系统性能。敏捷开发平台除了参数配置外, 还支持通过系统配置生成 JSP 代码文件, 这样, 大部分配置通过生成 JSP 文件实现, 少量的系统调整通过参数配置来实现, 系统结构非常清晰^[3]。

敏捷开发平台配置原理如图 1 所示。

1.2 敏捷开发平台的系统架构

平台化已经成为管理信息系统发展趋势, 这样的平台一般都是由很多组件构成, 技术实现的核心是一个引擎, 用来支持组件的运行, 这样的平台的稳定性依赖于引擎的好坏, 平台的不稳定可能会给项目造成很大影响。另外, 引擎式平台的另外一个缺陷是组件的兼容性差, 组件无法

收稿日期: 2011-09-23; 修订日期: 2012-04-12

作者简介: 李新 (1966-), 男, 山东淄博人, 博士, 副研究员, 研究方向为软件工程。E-mail: ca365@sohu.com

在不同平台之间复用, 这给系统在不同平台之间迁移带来很大困难^[4-5]。

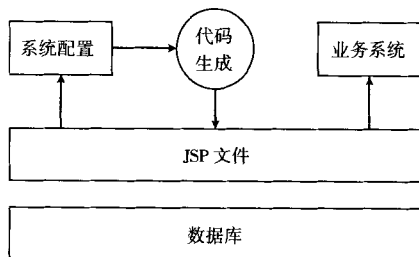


图1 敏捷开发平台配置原理

敏捷平台的核心技术是根据组件的配置生成 JSP 代码, JSP 文件直接运行在 Web 服务器上, 没有组件引擎, 不会产生因为引擎设计问题而带来的系统的不稳定问题。生成的 JSP 可以复制到任何支持 Java 的 Web 服务器上运行, 代码的复用率高。

敏捷平台除了“敏捷元素”、“敏捷表单”、“敏捷 workflow”、“敏捷查询”、“敏捷组件”等内容外, 还包括一套完整的“敏捷系统框架”, 这套框架支持机构、菜单、用户、角色、权限等管理模块, 能够满足一般管理信息化系统的要求。

敏捷平台系统架构如图2所示。

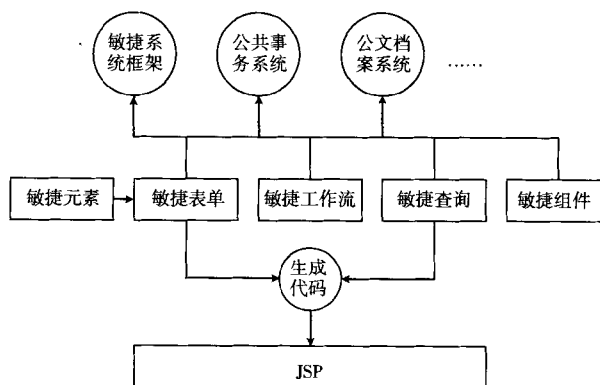


图2 敏捷平台系统架构

1.3 敏捷组件实现原理

敏捷开发平台的核心是一系列敏捷组件, 敏捷组件的核心是代码生成器, 即根据组件的配置信息, 生成相应的 JSP 代码文件, 由这些 JSP 文件支撑相应的业务功能。

敏捷开发平台将常用的业务逻辑提炼成标准的业务逻辑, 然后制作相应的程序模板, 在开发过程中根据业务需要, 配置敏捷组件, 敏捷组件的配置信息保存在数据库中, 代码生成器读取程序模板和组件的配置信息, 生成符合要求的 JSP 文件。

敏捷组件配置——生成文件如图3所示。

对于不适合生成 JSP 代码的业务逻辑, 敏捷组件可以在运行时根据业务的要求从数据库中读取配置, 实现动态

配置功能。

敏捷组件配置——读取配置如图4所示。

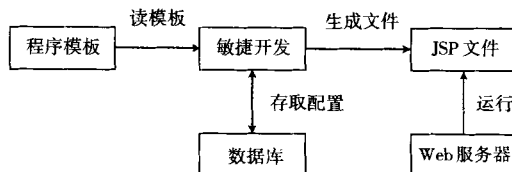


图3 敏捷组件配置——生成文件

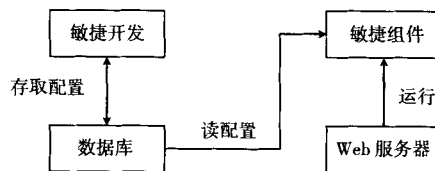


图4 敏捷组件配置——读取配置

1.4 敏捷查询、表单及数据表的设计

信息化系统的主要工作可以归结为对数据库表的增、删、改、查。敏捷开发平台的数据库表要经过规范化设计, 每个表必须具备一个 ID 字段, 还必须有一个 CLOB 字段, 用于存储 XML。读数据时, 先将数据库字段打包成 XML 文件, 然后将 XML 传到客户端, 用户的客户端界面主要是一个 HTML 文件模板, 接收到 XML 数据后将数据解析, 然后匹配到模板中相应的区域。写数据时, 先将数据打包成 XML 文件, 然后将 XML 传到服务器保存到数据库表的 XML 字段, 同时解析并保存到其它字段。

由于有 XML 这个字段, 从理论上来说任何数据都可以存在一个相同的表中, 不需要另外设计表结构。特别是在用户需求不是十分明确的需求调研阶段, 采用敏捷开发平台, 只需要配置相应的界面模板, 就可以实现数据的简单管理, 这在开发系统原型是特别有用。当用户需求稳定后, 通过一个自动生成程序, 生成不同的表结构, 将不同数据保存到不同的数据表中。

敏捷表单工作原理如图5所示。

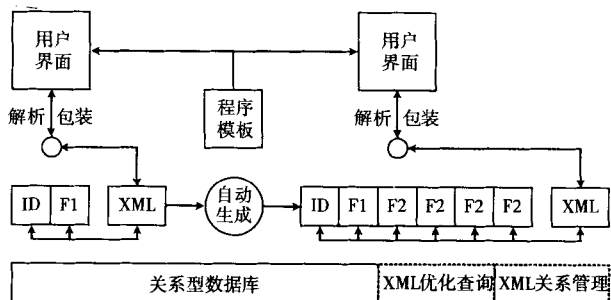


图5 敏捷表单工作原理

2 敏捷 workflow 及数据交换平台

除了组件这种细粒度的程序模块外, 一个成熟的开发

平台, 还应该提供 workflow、数据交换平台这种常用的业务功能模块。不同于第三方的 workflow 系统和数据交换平台, 敏捷开发平台提供的这两个业务模块与敏捷组件紧密集成, 使用起来更加简单方便。

2.1 敏捷 workflow

流程管理是信息化系统的常见业务, 为解决各种复杂的数据流向, 很多商业化的 workflow 引擎软件需要编写复杂的程序, 二次开发的工作量较大。实际上, 信息化系统中的流程一般可以简化为一条直线 (例如: 审批过程一般是由低向高传递), 分支和循环可以通过在直线流程上增加跳转、返回、选择功能来实现^[6-7]。

敏捷 workflow 由若干个相互连接的环节组成, 这些环节是预先定义好的, 每个环节都可以定义不同的成员, 当流程流转到相应环节时, 只有环节中的成员可以处理相应业务。每个环节都可以定义不同的页面, 用于对相应业务做不同处理。

预定义环节预设了流程的步骤及每个环节的参与人员, 在每个预定义环节中, 用户还可根据业务需要临时自定义一些环节。敏捷 workflow 通过支持多种人员选择方式、流程跳转约束、流程批处理等方式, 实现复杂的流程控制。

为了处理流程并行的情况, 在主控流程的基础上, 可以定义若干个并行流程, 每个并行流程可以单独定义自己的业务逻辑及用户界面。

敏捷 workflow 如图 6 所示。

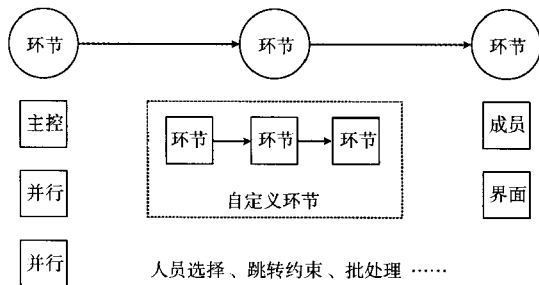


图 6 敏捷 workflow

2.2 敏捷数据交换平台

对于大型的跨地域的信息系统, 需要解决跨部门、跨组织的应用之间的数据交换、数据共享、信息流转问题。数据交换平台的目的是在异构环境中实现系统之间的数据流通和共享, 从而有效地利用资源, 方便系统集成^[8-9]。

敏捷开发平台采用基于 MQ (message queue) 消息中间件的数据交换机制。消息中间件位于客户机/服务器的操作系统之上, 管理计算资源和网络通讯。

消息中间件利用高效可靠的消息传递机制进行平台无关的数据交流, 并在应用程序之间传送消息, 这些消息可以在不同的网络协议、不同的计算机系统和不同的应用软件之间传递。

敏捷数据交换平台系统由一台中心服务器和多台节点工作站组成。所有消息都要先经过中心服务器的中转, 才能最终转换到节点工作站中, 形成了一个星型的网络拓扑结构。如果采用网状拓扑结构, 比较难以维护, 而采用这种星型拓扑结构, 可以有效地隔离各个节点的故障, 并且易于管理, 而作为用户的节点工作站, 并不知道中心服务器的存在, 感觉和两两直接连接一样。

在各个节点工作站中有 3 个存储介质: Outbox、Inbox、Log。Outbox 用于存储发送出去的数据, Inbox 用于存储接受到的数据, Log 作为一个控制期, 主要用于记录每次操作的日志。3 个存储介质采用了消息队列的方式存储数据。

敏捷数据交换平台的架构如图 7 所示。

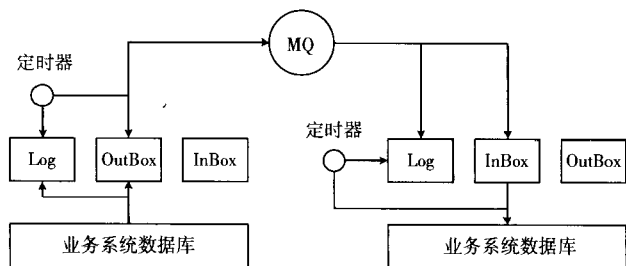


图 7 敏捷数据交换平台架构

3 敏捷开发的数据库设计

随着信息技术的发展, 数据资源已经成为信息系统的核心。良好的数据库设计必须达到数据结构清晰、稳定的要求, 并且随着系统功能的演进和软件版本的升级, 数据库结构要保持相对稳定。

3.1 数据资源的规划设计

敏捷开发平台将数据资源进行统一规划, 建立数据资源目录。数据资源目录呈树型结构, 目录树的节点之间不仅能实现父子关系, 还要能够相互引用, 以实现复杂的分类关系。如图 8 所示, 在数据资源目录中, 节点 5 是节点 2 的子节点, 但同时被节点 3 以子节点方式引用, 也是节点 3 的子节点, 也就是说, 节点 5 及以下 8、9 两个节点, 同时隶属于节点 2、3 两个分类。

关系型数据库采用主外、键关系来表达数据结构, 大型信息化系统数据结构非常复杂, 这种“点线关系”会形成复杂的“网状结构”, 如果数据表比较多, 形成的复杂关系网常常令人难以理解。敏捷开发平台将关系型数据库中的多对多的关系统一到一张关系表中 (表: agiRel), 在表 agiRel 中, 字段 TNAMEA 表示 A 表表名, 字段 TNAMEB 表示 B 表表名, 字段 IDA 表示 A 表主键, 字段 IDB 表示 B 表主键, 这样就将复杂的网状结构简化成一种“总线结构”。

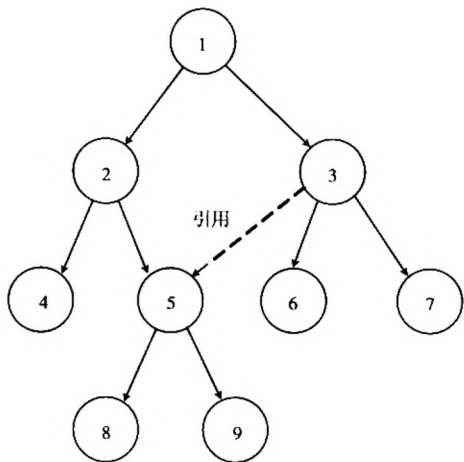


图 8 数据资源目录结构

3.2 数据资源的存储设计

敏捷开发平台采用元数据的方式建立业务数据表，相同结构或类似结构的数据存储在同一张物理表里。即同一张数据表中的数据只是结构相同，业务之间可能毫无关系，为表达数据之间的业务关系，需要在物理表上建立不同的业务视图。

敏捷开发平台的基本数据结构如图 9 所示，表 agiDir 存储数据资源目录，agiRel 存储多对多关系，agiLob 存储 CLOB 或 BLOB 类数据（比如：各种图片、媒体及附件文件），agiData 存储某种结构的业务数据。系统中类似 agiData 的数据表可能会有多个，将数据存储在多个数据表中的原因一方面是数据结构的差异，另一方面是为了在大数据量的情况下提高数据存取效率。

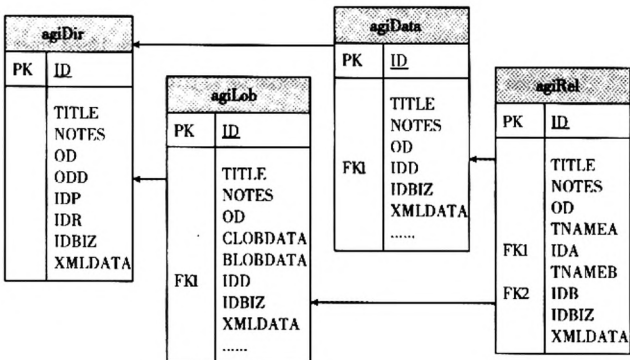


图 9 敏捷开发平台数据库基本结构

在实际系统设计中，非结构化数据都存储在 agiLob 中；结构化数据可存储在单独的 agiData 表里，也可以存储在 agiDir 的 XMLDATA 字段中。agiData 表的创建、删除及数据在 agiDir 和 agiData 中的转移可以通过程序自动实现。这样，在需要的情况下，数据库可以收缩在 agiDir、agiLob 和 agiRel 这 3 张表中，数据的迁移和系统升级会变得十分容易。

3.3 数据库表设计规范

首先是命名规则，考虑英文非中国人母语，宜采用对象名称的汉语拼音首字母来命名。如果有两个对象的拼音首字母重复，可将重复的部分取完整声母、韵母或全拼加以区别。对于常用的英文缩写，可编成英文缩写词汇表，词汇表应当尽量简洁，在项目中严禁使用词汇表以外的英文词汇。常用的英文缩写可参考表 1。

表 1 英文缩写词汇

缩写	含义	全拼
ADDR	地址	address
ATTR	属性	attribute
CN	中国	china
DEPT	部门	department
EMAIL	电子邮件	email
FAX	传真	fax
FEE	费用	fee
INFO	信息	information
LAB	实验室	laboratory
NO	号码	number
SEX	性别	sex
SRC	来源	source
TEL	电话	tel
VER	版本	version

另外，需要对字段的设计进行规范。以 Oracle 为例，允许使用的数据类型为 VARCHAR2、CLOB、BLOB、DATE、NUMBER。VARCHAR2 类型的字段长度按照向上取整的方法采用以下序列值：50、500、4000，缺省值为零长度字符串。一般不用布尔类型字段，需要时用整数代替（0 表示 false、1 表示 true）。NUMBER 类型的数据精度与小数位数，应根据系统的实际情况做规范，对于不涉及对数据特殊要求的项目，一般整数用 NUMBER（10），小数用 NUMBER，人民币用 NUMBER（，2）。

每一张数据表中必须有 ID、Title、NOTES 等字段，其含义见表 2。

表 2 数据表中必须包含的字段

名称	描述	类型	长度	是否必填
ID	表里的唯一主键，是数据表的主键，基本不涉及业务。	VARCHAR2	50	是
TITLE	数据名称	VARCHAR2	500	否
NOTES	数据的描述	VARCHAR2	4000	否
OD	序号，主要用于显示界面顺序。	Number		否
XMLDATA		CLOB		否
IDBIZ	业务主键，表示数据的唯一性，有实际意义的 ID。			

4 敏捷开发平台的优势

对于敏捷,不同的人有着不同的理解,敏捷的核心价值和目的是“适配性”,基于敏捷思想的系统应该是可搭建、可配置的系统。因此,敏捷开发平台着重解决了以下几个问题:

(1) 系统设计标准化:敏捷开发平台通过规范化的程序模板和数据库结构,简化了系统设计,降低了系统开发的难度和随意性,保证了软件工期和程序质量,其清晰的数据库结构方便了系统升级、维护和数据利用^[10]。

(2) 数据的可调整:在系统开发过程中经常遇到需要改变数据结构的情况,这一方面是由于在需求阶段很难完全正确把握业务需求,另一方面,随着业务的发展和认识的深入,原来的系统设计可能无法满足新的需求。因此,要求系统能够在修改数据结构时不会对系统产生很大影响。敏捷开发平台做到了数据结构可配置,在调整数据结构时不必重写代码。

(3) 流程的可配置:信息化的目的之一就是促进管理创新,业务流程的变更是管理创新的最常见表现形式。敏捷开发平台包含一套完整的工作流系统,工作流是动态调整的,业务流程的调整可在界面进行,不必写程序。

(4) 支持在线更新:敏捷开发平台全部程序代码都保存在平台的数据库中,系统正在运行的都是最新的程序代码,克服了传统的编译方式带来的程序代码版本管理问题,具备热部署及在线更新的功能,在不停机的情况下可以方便地完成系统更新。

(5) 业务流程跨系统:在我国,一些大型的企事业单位往往采用总部、地区分中心形式的多级组织机构管理体系,一些业务过程需要逐级审批流程。敏捷工作流通过敏捷数据交换平台,实现了跨系统的业务流程,可以在分布式部署的系统中,实现集中式系统的用户体验。

5 结束语

由于大型信息化项目迭代开发的成本太高,一般认为,敏捷开发只适合中小型项目。有了敏捷开发平台的支持,业务过程、用户界面、程序实现都可以按照一种标准化的方法来处理,信息系统的开发难度有效降低,这使得大型信息化项目也可以采用敏捷开发方法,从而更好的适应用户业务需求的变化。

敏捷开发平台已经成功应用于中国科学院资源规划系统、国家自然科学基金委信息化系统这样的特大型信息化项目,取得了良好效果。

参考文献:

[1] YANG Ping, ZHOU Yuncheng, GUO Dan, et al. Analysis and comparison of processing models in software engineering sector [J].

Journal of Jixi University, 2008, 8 (1): 86-88 (in Chinese). [杨萍, 周云成, 郭丹, 等. 软件工程中主要开发模型的分析 [J]. 鸡西大学学报, 2008, 8 (1): 86-88.]

- [2] SONG Chengji, CHEN Xiaojian. Extreme programming and its application to the development of administrative system of the buying and selling [J]. Computer Application and Software, 2008, 25 (9): 4342-4344 (in Chinese). [宋承继, 陈小健. 极限编程在进销存管理系统开发中的应用 [J]. 计算机应用与软件, 2008, 25 (9): 4342-4344.]
- [3] ZHAO Yuehua, WANG Ling. Design of Java code generation based on agile [J]. Computer Engineering and Design, 2009, 30 (12): 3018-3021 (in Chinese). [赵跃华, 王凌. 基于敏捷方式的 Java 代码生成方法的设计 [J]. 计算机工程与设计, 2009, 30 (12): 3018-3021.]
- [4] ZHANG Zhuxi, LI Renjie, WANG Huaimin. One trusted software constitution environment based on aspect-oriented programming [J]. Application Research of Computers, 2009, 26 (5): 1743-1745 (in Chinese). [张曷熹, 李仁杰, 王怀民. 一个面向方面的可信软件开发平台 TSCE [J]. 计算机应用研究, 2009, 26 (5): 1743-1745.]
- [5] DENG Zhidong, YU Shiliang, CHENG Zhenbo. General-purpose distributed virtual reality development platform based on high level architecture [J]. Journal of System Simulation, 2008, 20 (12): 3160-3164 (in Chinese). [邓志东, 余士良, 程振波. 通用分布式虚拟现实软件开发平台的究 [J]. 系统仿真学报, 2008, 20 (12): 3160-3164.]
- [6] LI Haibo, ZHAN Dechen, XU Xiaofei. Architecture of component composition based on workflow engine [J]. Journal of Software, 2006, 17 (6): 1401-1410 (in Chinese). [李海波, 战德臣, 徐晓飞. 基于工作流引擎的构件组装体系结构 [J]. 软件学报, 2006, 17 (6): 1401-1410.]
- [7] LIU Min, YAN Junwei, WANG Jian. WS-resource based inter-operation framework for dynamic workflow [J]. Computer Integrated Manufacturing System, 2006, 12 (2): 264-270 (in Chinese). [刘敏, 严隽薇, 王坚. 基于 Web 服务与资源模型的动态工作流互操作框架 [J]. 计算机集成制造系统, 2006, 12 (2): 264-270.]
- [8] MEI Lijun, FU Xiaolong, LIU Qixin, et al. Research and implementation of data-exchange platform based on SOA [J]. Computer Engineering and Design, 2006, 27 (19): 3601-3603 (in Chinese). [梅立军, 付小龙, 刘启新, 等. 基于 SOA 的数据交换平台研究与实现 [J]. 计算机工程与设计, 2006, 27 (19): 3601-3603.]
- [9] LIU Yanhui, DONG Bidan, ZHANG Feng. Applied research of data-exchange platform for distributed [J]. Computer Engineering and Design, 2009, 30 (16): 3780-3782 (in Chinese). [刘艳辉, 董碧丹, 张峰. 数据交换平台的分布式应用研究 [J]. 计算机工程与设计, 2009, 30 (16): 3780-3782.]
- [10] Lixin. Study of software engineering based on normalization [J]. Journal of Computer Applications, 2005, 26 (Z2): 228-231 (in Chinese). [李新. 基于标准化的软件工程学研究 [J]. 计算机应用, 2005, 26 (Z2): 228-231.]