

软件过程模型选型 说明书

课程名称：_____系统分析与设计_____

任课教师：_____刘 波_____

专 业：_____软件工程_____

2020 年 3 月 26 日

目 录

一、 项目名称及概要	1
1.1 项目名称	1
1.2 概要	1
二、 团队信息与分工说明	1
2.1 团队信息	1
2.2 分工说明	2
三、 项目特征和团队背景分析	2
3.1 项目特征	2
3.2 团队背景	3
四、 候选软件过程模型	3
4.1 瀑布模型	3
4.2 喷泉模型	5
4.3 可重用部件模型	6
4.4 螺旋模型	7
4.5 RAD 模型	9
4.6 SCRUM 模型	12
五、 软件过程模型选型结论	14
5.1 选型结论：RAD 模型	14
5.2 原因阐述	14
六、 参考文献	15

一、项目名称及概要

1.1 项目名称

西大服务平台—学生服务系统

1.2 概要

1.2.1 背景

西南大学的学生在进行重修选课,查看课表以及成绩时需要频繁使用学校的教务系统。目前西南大学内的教务系统的需要检测学生的 IP 地址,只有属于学校内的 IP 地址才能拥有登录学校内部教务系统的权限。在实际中由于各种原因学生无法连接学校的内部网络,但是仍然有使用学校教务系统的需求,学生只能通过 VPN 的方式来访问学校内网。由于 VPN 的信号不稳定,而且查询过程非常复杂,使用这种方式来进行查询会浪费许多不必要的时间。西南大学的学生需要一个拥有便捷查询功能的系统来满足教务系统简单的查询需求。

1.2.2 功能目标

- (1) 在首次使用功能需要绑定学生学号;
- (2) 按周查看学生课表;
- (3) 按学期查看成绩排名和各科绩点;
- (4) 导出标准教务系统成绩单;
- (5) 查看教室预约情况;
- (6) 查看考试时间。

1.2.3 工作内容

在本项目的开发中须进行的主要工作:进行可行性研究、制定项目开发计划、进行需求分析、进行概要设计、进行详细设计、进行用户界面设计、编码、测试、编写文档、运行与维护、按计划提请阶段评审、提交测试人员评测开发产品、交付最终工作产品、项目实施总结、项目验收。

二、团队信息与分工说明

2.1 团队信息

(按学号顺序)

222018321062004

张津豪

222018321062005	刘 航
222018321062006	宋行健
222018321062007	范静鸿
222018321062060	陈禹桐
222018321062061	刘怡锦

2.2 分工说明

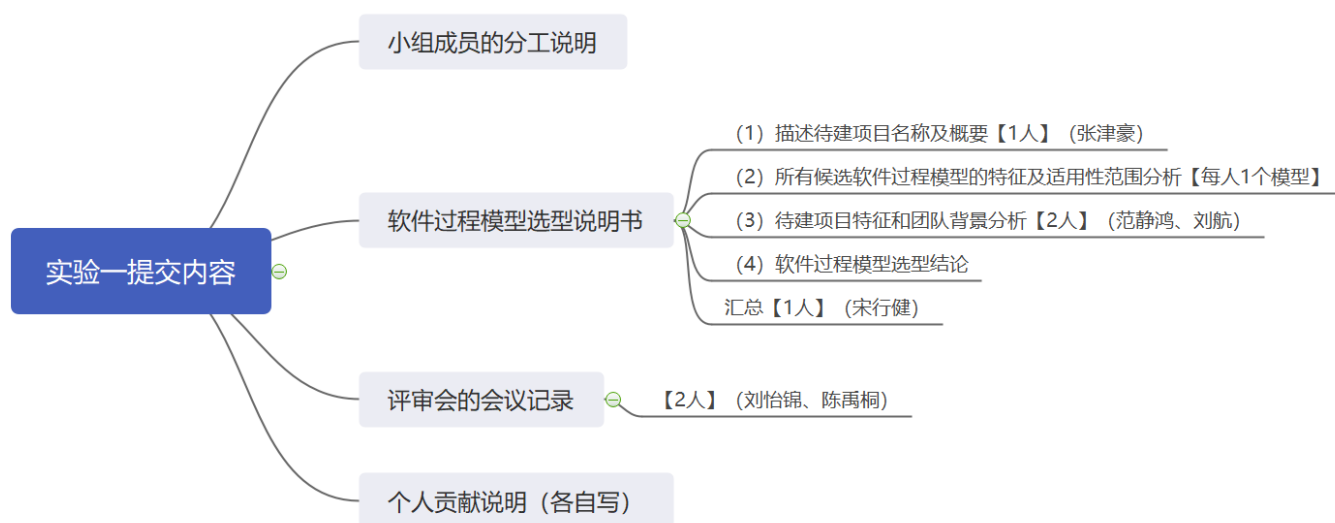


图 1 分工说明

三、项目特征和团队背景分析

3.1 项目特征

3.1.1 系统结构

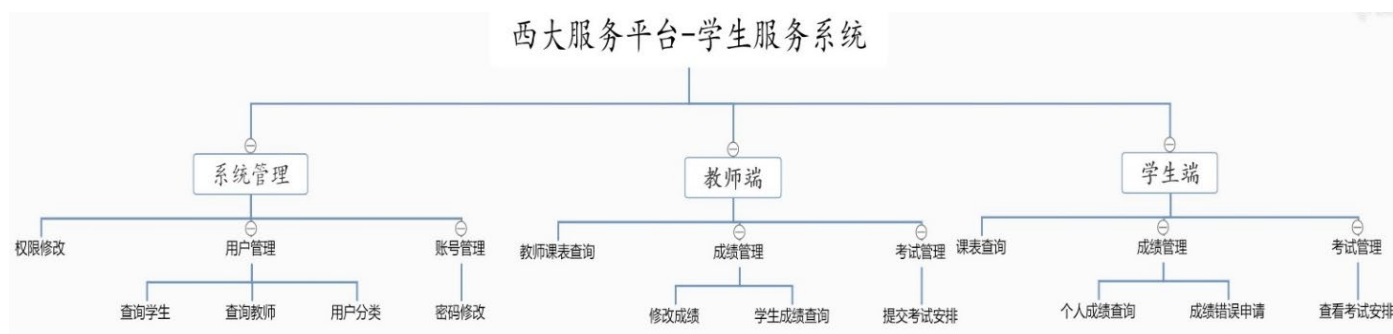


图 2 系统结构图

“西大服务平台—学生服务系统”是一个以查询为主要功能的一站式服务系统，提供对学生用户移动端的成绩、课表等一系列内容的快捷查询服务。

- (1) 需求明确且比较固定，需求变化不剧烈；
- (2) 项目工期紧，但是迭代周期少，项目完成后变化不大；
- (3) 数据要共享，避免信息孤岛；
- (4) 每项功能的需要的数据量大，属于数据密集型项目；
- (5) 各项功能之间可以拆分，模块化；
- (6) 基于学校教务系统的功能，使用学校数据库；
- (7) 账户与用户（学生）一对一；
- (8) 通过模块化的形式使用户能通过点击接口按钮跳转相应板块；
- (9) 项目功能较简单且互相独立（查询、导出），非大型项目。

3.2 团队背景

- (1) 团队成员总数为 6 人，属于人数偏少的团队。男女均有，逻辑思维与情感设计能力兼备，能够更全面分析用户需求，设计更好的项目；
- (2) 开发人员均为软件工程专业学生，有强大的代码编写能力，可以较好的完成底层架构；
- (3) 开发人员同时也是项目目标用户，对于用户需求非常了解，测试比较便捷；
- (4) 项目经理具有强大的组织能力与逻辑思维，能够很好地进行项目分工与风险管控，减少项目的错误；
- (5) 团队资金不多，设备有限，不能长时间开发。

四、候选软件过程模型

4.1 瀑布模型

4.1.1 简介

瀑布模型（Waterfall Model）是一个项目开发架构，开发过程是通过设计一系列阶段顺序展开的，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈，因此，如果有信息未被覆盖或者发现了问题，就“返回”上一个阶段并进行适当的修改。

项目开发的整个进程是从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。

4.1.2 核心思想

瀑布模型核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现与物理实现分开。将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。

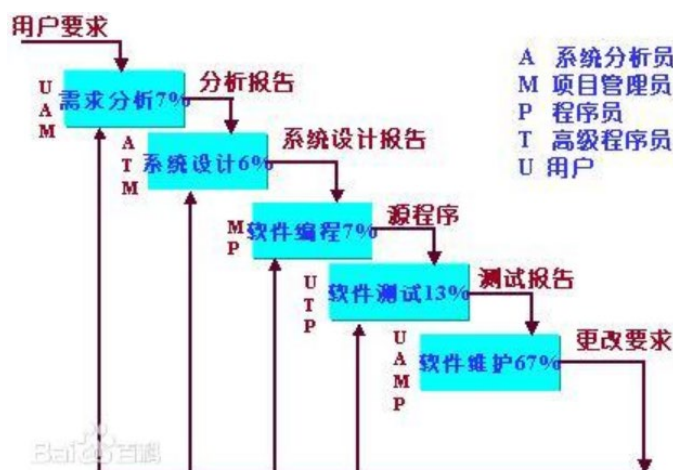


图 3 瀑布模型示意图

4.1.3 优势

- (1) 为项目提供了按阶段划分的检查点；
- (2) 当前一阶段完成后，您只需要去关注后续阶段；
- (3) 可在迭代模型中应用瀑布模型；
- (4) 它提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。

4.1.4 缺点

- (1) 各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量；
- (2) 由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发风险；
- (3) 早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果
- (4) 通过过多的强制完成日期和里程碑来跟踪各个项目阶段；
- (5) 不应用户需求的变化。

4.1.5 模型适用范围

瀑布模型适合于结构化方法，也就是面向过程的软件开发方法。

在瀑布模型中，软件开发的各项活动严格按照线性方式进行，当前活动接受上一项活动的工作结果，实施完成所需的工作内容。当前活动的工作结果需要进行验证，如果验证通过，则该结果作为下一项活动的输入，继续进行下一项活动，否则返回修改。

瀑布模型强调文档的作用，并要求每个阶段都要仔细验证，对于经常变化的项目而言，瀑布模型毫无价值。

软件项目或产品选择瀑布模型必须满足下列条件：在开发时间内需求没有或很少变化；分析设计人员应对应用领域很熟悉；低风险项目（对目标、环境很熟悉）；用户使用环境很稳定；用户除提出需求以外，很少参与开发工作。

4.2 喷泉模型

4.2.1 简介

喷泉模型（fountain model）是一种以用户需求为动力，以对象为驱动力的模型，主要用于描述面向对象的软件开发过程。该模型认为软件开发过程自下而上周期的各阶段是相互迭代和无间隙的特性。

4.2.2 特征

布式特点当软件结束计划阶段，分布在不同地域的软件开发小组可以根据计划，在不同或者相同的时间启动项目开发。

测试的充分软件测试中测试用例的覆盖率直接决定了软件测试的质量。改进的喷泉模型大大扩大了设计和选取测试用例的范围，可以从包括程序、文档等所有可以使用的信息中获得，提高了测试用例的覆盖率，保证测试的充分性和完全性。

完全实现了测试和开发的同步，以及各个过程内各个阶段之间的同步。真正实现了“全过程”测试，提高了软件测试的质量。

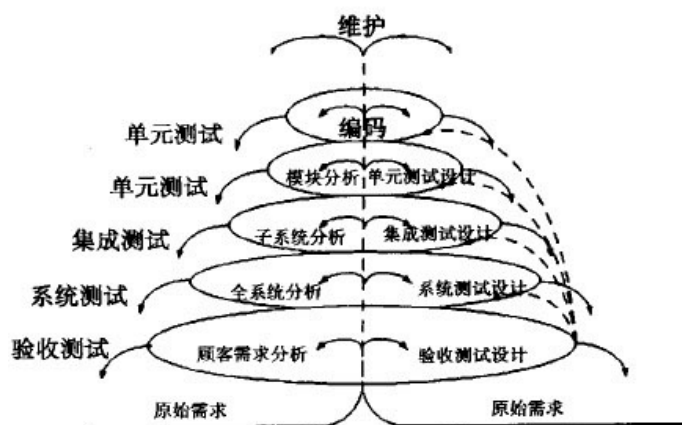


图 4 改进的喷泉模型

4.2.3 优势

喷泉模型不像瀑布模型那样，需要分析活动结束后才开始设计活动，设计活动结束后才开始编码活动。该模型的各个阶段没有明显的界限，开发人员可以同步进行开发。其优点是可以提高软件项目开发效率，节省开发时间，适应于面向对象的软件开发过程。

4.2.4 缺点

由于喷泉模型在各个开发阶段是重叠的，因此在开发过程中需要大量的开发人员，因此不利于项目的管理。此外这种模型要求严格管理文档，使得审核的难度加大，尤其是面对可能随时加入各种信息、需求与资料的情况。

4.3 可重用部件模型

4.3.1 简介

对象技术为软件工程的基于构件的过程模型提供了技术框架。面向对象范型强调了类的创建，类封装了数据和用于操纵该数据的算法。如果经过合适的设计和实现，面向对象的类可以在不同的应用及基于计算机的系统结构中复用。构件组装模型融合了螺旋模型的许多特征。它本质上是演化的支持软件开发的迭代方法。但是，构件组装模型是利用预先包装好的软件构件（也称为“类”）来构造应用程序的。

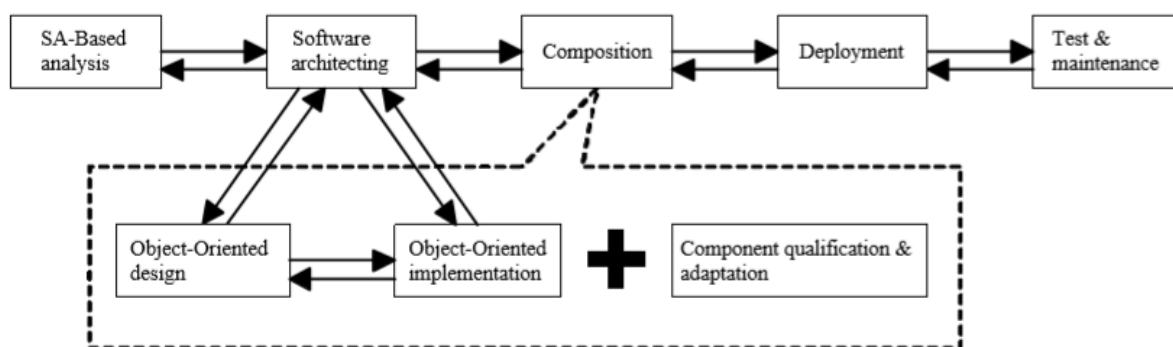


图 5 可重用构件组装的标准开发流程^[5]

4.3.2 特征

随着基于 CBSD 理论日趋成熟，对构件组装机制的研究日益深入，出现了两种典型的构件组装机制——结合中间件技术的构件组装和基于进程的构件组装。^[4] 中间件技术对构件描述的抽象层次比较低，组装时有先天性的缺陷，构件间的关系表述不直观。构件间的耦合关系过于紧密；构件组装描述和构件运行环境绑定在一块，不利于异种构件的集成。构件具有可独立部署的特性，因此也可将构件看作服务或进程。这种方式对构件组装描述的抽象层次较高，但是，构件本身并不等同于通常意义下的进程，数量大时，系统开销过大，组装效率低下。

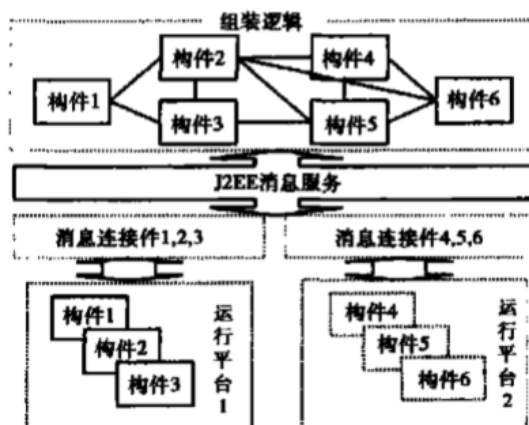


图 6 多个平台上的构件连接

4.3.3 优势

- (1) 大大提高了软件开发的效率;
- (2) 而且构件模型允许多个项目同时开发, 降低了费用, 提高了可维护性。

4.3.4 缺点

- (1) 由于存在多种构件标准, 缺乏通用的构件组装结构标准, 如果自行定义标准的话, 会引入较大的风险;
- (2) 构件可重用性和软件系统高效性之间不易协调, 需要权衡。

4.3.5 适用性

本系统的功能之间分工比较明确, 每一项功能可以单独进行开发。在本次开发中我们的团队规模较小, 很容易统一开发标准。各功能分开的开发流程也比较符合目标的需求

但是由于本系统中各功能都依赖于同一个教务系统, 如果按照构建开发可能会产生代码的冗余从而影响性能。同时本开发方式要求各构件开发成员有独立开发系统的能力, 对代码能力不高的开发成员来说比较难以进行。

4.4 螺旋模型

4.4.1 简介

螺旋模型是一种演化软件开发过程模型, 它兼顾了快速原型的迭代的特征以及瀑布模型的系统化与严格监控。螺旋模型最大的特点在于引入了其他模型不具备的风险分析, 使软件在无法排除重大风险时有机会停止, 以减小损失。同时, 在每个迭代阶段构建原型是螺旋模型用以减小风险的途径。螺旋模型更适合大型的昂贵的系统级的软件应用。

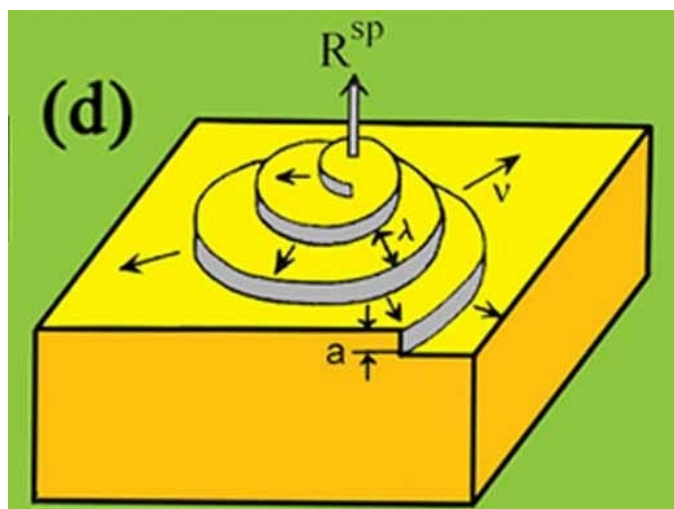


图 7 螺旋模型示意图

4.4.2 特征



图 8 螺旋模型过程示意图

螺旋模型(Spiral Model)采用一种周期性的方法来进行系统开发。这会导致开发出众多的中间版本。使用它,项目经理在早期就能够为客户实证某些概念。该模型是快速原型法,以进化的开发方式为中心,在每个项目阶段使用瀑布模型法。这种模型的每一个周期都包括需求定义、风险分析、工程实现和评审 4 个阶段,由这 4 个阶段进行迭代。软件开发过程每迭代一次,软件开发又前进一个层次。软件过程螺旋模型基本做法是在"瀑布模型"的每一个开发阶段前引入一个非常严格的风险识别、风险分析和风险控制,它把软件项目分解成一个个小项目。每个小项目都标识一个或多个主要风险,直到所有的主要风险因素都被确定。

螺旋模型沿着螺线进行若干次迭代，图中的四个象限代表了以下活动：

- (1) **制定计划:** 确定软件目标, 选定实施方案, 弄清项目开发的限制条件;
- (2) **风险分析:** 分析评估所选方案, 考虑如何识别和消除风险;
- (3) **实施工程:** 实施软件开发和验证;
- (4) **客户评估:** 评价开发工作, 提出修正建议, 制定下一步计划。

螺旋模型由风险驱动，强调可选方案和约束条件从而支持软件的重用，有助于将软件质量作为特殊目标融入产品开发之中。

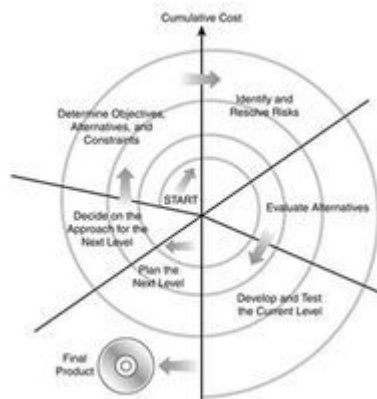


图 9 螺旋示意图

4.4.3 优势

- (1) 设计上的灵活性,可以在项目的各个阶段进行变更。
- (2) 以小的分段来构建大型系统,使成本计算变得简单容易。
- (3) 客户始终参与每个阶段的开发,保证了项目不偏离正确方向以及项目的可控性。
- (4) 随着项目推进,客户始终掌握项目的最新信息,从而他或她能够和管理层有效地交互。
- (5) 客户认可这种公司内部的开发方式带来的良好的沟通和高质量的产品。

4.4.4 缺点

- (1) 很难让用户确信这种演化方法的结果是可以控制的。
- (2) 建设周期长,而软件技术发展比较快,所以经常出现软件开发完毕后,和当前的技术水平有了较大的差距,无法满足当前用户需求。

4.4.5 适用性

- (1) 螺旋模型强调风险分析,但要求许多客户接受和相信这种分析,并做出相关反应是不容易的,因此,这种模型往往适应于内部的大规模软件开发。

- (2) 如果执行风险分析将大大影响项目的利润，那么进行风险分析毫无意义，因此，螺旋模型只适合于大规模软件项目。
- (3) 软件开发人员应该擅长寻找可能的风险，准确地分析风险，否则将会带来更大的风险。对于新近开发，需求不明确的情况下，适合用螺旋模型进行开发，便于风险控制和需求变更。

4.5 RAD 模型

4.5.1 简介

快速应用开发（Rapid Application Development、RAD）是线性顺序开发模型（瀑布模型）的高速变种，强调极短的开发周期。该模型通过大量使用可复用构件，采用基于构件的建造方法获得快速开发。如果正确理解需求，且约束项目范围，采用 RAD 模型可以很快创建功能完善的信息系统。其目标是要缩短传统的 SDLC（System Development Life Cycle）方法中信息系统的设计与实现之间漫长的时间间隔，尽量更好地满足迅速变化的商业需求。

4.5.2 开发流程

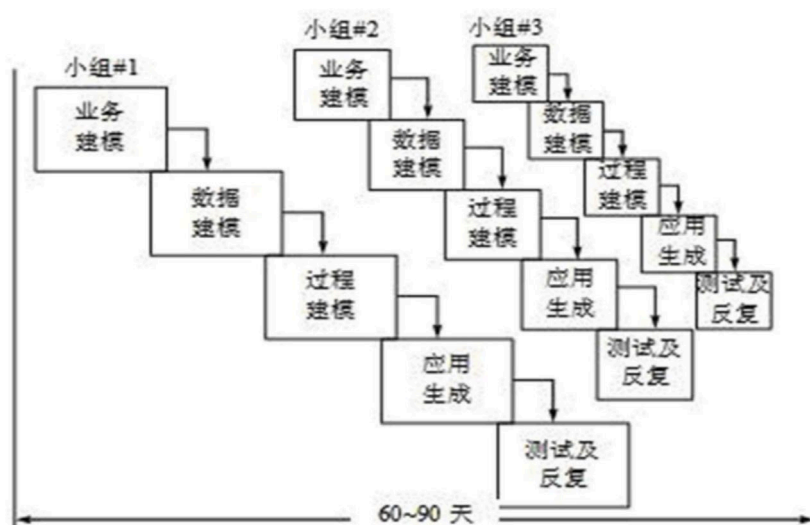


图 10 RAD 模型开发过程

RAD 模型流程是从业务建模开始随后是数据建模、过程建模、应用生成、测试及反复。其中：

1. **业务建模**：确定驱动业务过程运作的信息，要生成的信息，如何生成，信息流的去向及处理等，可以用数据流图（Data flow diagram，具体见图 11）辅助。在本项目中，运作目标是建立完善且简洁的教务系统相关功能，对应的使用用户是学生，教师及内部行政人员。

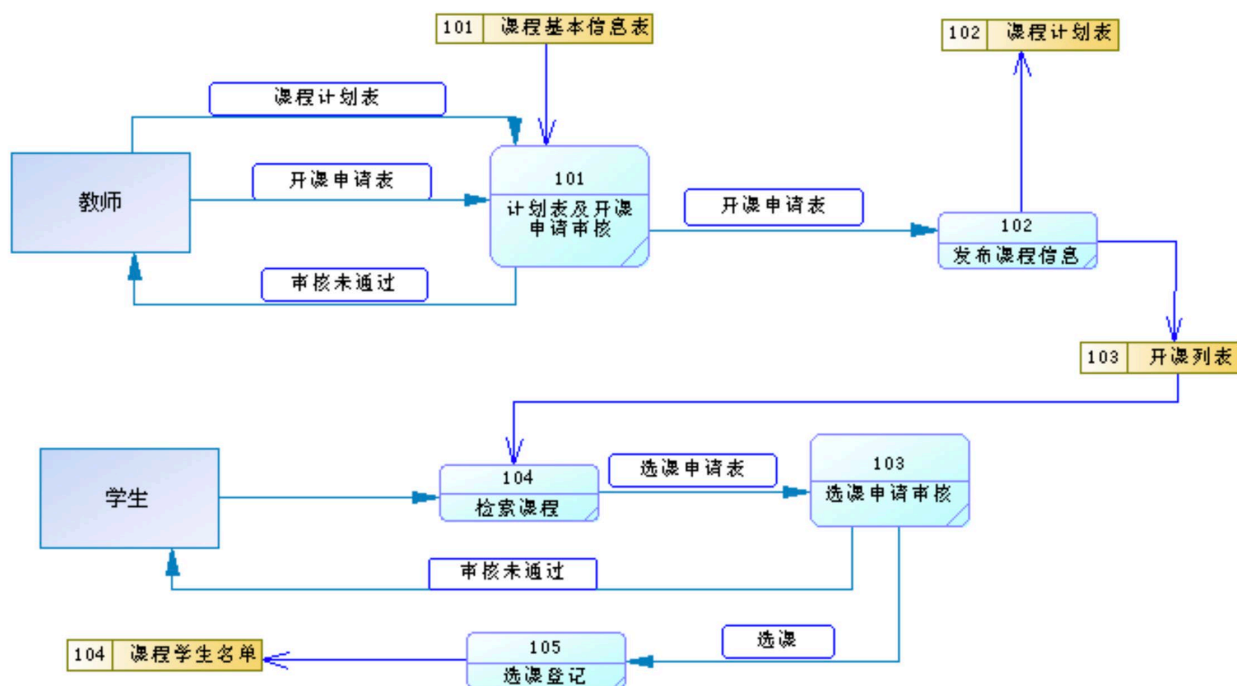


图 11 学生课表查询申请数据流图

2. 数据建模：为支持业务过程的数据流查询数据对象集合、定义数据对象属性，并与其他数据对象的关系构成数据模型，可以用 E-R 图（详见图 12）辅助。在本项目中，使用对象及属性是学生（学号，姓名，入学年份，课表，考试时间，成绩，教室），教师（教师号，教室，所教班级）

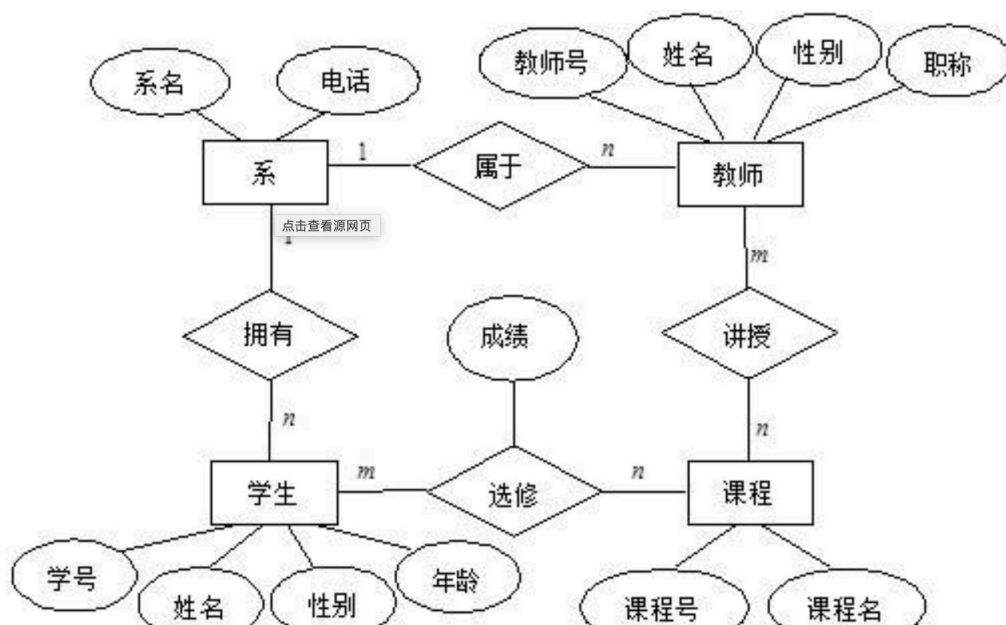


图 12 学生课程 E-R 图

3. 过程建模：使数据对象在信息流中完成各业务功能，创建过程来描述数据对象的增加、修改、删除、查找，即细化数据流图中的处理框。针对本项目，要具体将学生分为不同

学院，不同年级；教师分为不同岗位，不同科目。对于查询任务来说，当用户选择不同项目所对应的页面跳转应不相同。

4. **应用程序生成**：利用第 4 代语言（4GL, forth-generation language）写出处理程序，重用已有构件或创建新的可重用构件，利用环境提供的工具自动生成以构造出整个的应用系统。

5. **交付与测试**：一般来说做整体测试，新创建构件也需测试。

4.5.3 特征

RAD 模型不采用传统的第 3 代程序设计语言来创建软件，而是采用基于构件的开发方法复用已有的程序结构或使用可复用构件和或创建可复用的构件，运用第四代语言（4GL, forth-generation language）写出处理程序。在所有情况下，均使用自动化工具辅助软件创造。RAD 模型项目上的时间约束需要范围。当一个业务能够被模块化并使得其中每一个主要功能均可以短时间完成，则可选用 RAD。每一个主要功能可由一个单独的 RAD 组来实现，最后集成起来形成一个整体。这是在瀑布模型上的改进，并组合以下五个技术：

1. **进化原型(evolutionary prototyping)**，即创建一个健壮的原型，不断在此基础上按照需求的变化进行特征扩展和修改，直至原型系统达到最终系统的要求为止。在本项目中可以不断根据需求添加完善内容，具有较易维护的特点。

2. **CASE 工具 (Computer Aided (or Assisted) Software Engineering)**，可正向和反向进行工程，借助软件自带功能可大大减少工作量。

3. **RAD 开发小组**，在本组中成员均出自软件工程，具有良好的知识素养与批判性思维，在合作与讨论中推进工程的进行。

4. **交互式 JAD (Joint Application Development)**，利用 JAVA, C++等编程知识进行系统分析，适合项目不太复杂且专业知识过硬的小组。

5. **时间表**，利用时间表分板块规划出相应工作的时间阶段，提高整体效率。

4.5.4 优势

(1) **节省大量时间**：设计与实践间隔缩短，当确定好用户需求与项目范围时，项目实现的时间较短，对于内容不复杂的项目有效。

(2) **可以有效测试原型**：RAD 依赖于先前迭代的反应，当有足够的用户提供真实有效的反馈时，开发人员可从中获得足够的信息进行不断完善修改。

(3) **可复用**：针对已经拥有的软件原型，RAD 可在此基础上不断修改重复使用，大大减少项目的构建与设计，也可较为方便的完善工程。

4.5.5 缺点

(1) **人员数量**：对于大型项目而言，RAD 需要足够的人力，才能确保在规定时间内完成对应的板块，但对于本项目而言，工程总量不大，内容较为简单且清晰，因此可以考虑使用 RAD 模型。

(2) **需要开发小组各成员配合**：由于需要在很短的时间内完成需求分析与设计，任何一个小组的配合不当都会导致项目进度缓慢，因此需要各成员有较高的基础素养与配合能力，本小组人员较少，沟通方便且专业素质过硬，可以相对解决此困难。

- (3) **并非所有系统都合适**: 不能合理模块化的系统、高性能需求并且要调整构件接口的系统均不适合, 本项目模块化清晰且性能需求不高, 适用于 RAD 模型。
- (4) **只能用于信息系统开发, 不适合技术风险很高的情况**: 当一个新应用要采用很多新技术或当新软件要求与已有的计算机程序的高互操作性时, RAD 缺陷就会发生。本项目属于信息系统且风险不高, 适用于 RAD 模型。

4.5.6 适用性

总体来说 RAD 模型是在瀑布模型基础上发展而来, 克服瀑布模型的缺点, 减少由于软件需求不明确带来的开发风险, 使用不断建模反复测试的步骤将项目过程所需要的时间尽量减少化, 具体适用于以下几个方面:

- (1) 有较高模块化, 性能需求不高不需调整构件接口项目;
- (2) 信息系统技术风险低的项目;
- (3) 人员数量大且配合相对默契的项目;
- (4) 有健壮原型且较易维护的项目;
- (5) 有电脑现成工具可以借助的项目;
- (6) 小组成员专业素质过硬的项目;
- (7) 需要短时间见证成果, 内容不过于复杂的项目。

4.6 Scrum 模型

4.6.1 简介

Scrum 是迭代式增量软件开发过程, 通常用于敏捷软件开发。在 Scrum 敏捷开发中, 软件项目在构建初期被切分成多个子项目 (冲刺清单, **Sprint Backlog**), 各个子项目的成果都经过测试, 具备可视、可集成、可运行使用的特征。^[1] 每个子项目都是一个“冲刺”(Sprint), 在这个周期中, 有“每日站立会”来进行当天工作进度的总结, 和“冲刺燃尽图”来展示团队中每个成员的工作进度。

Scrum 开发中有三个成员角色:

- (1) **Product Owner (产品负责人)**: 主要负责确定产品的功能和达到要求的标准, 代表了客户的意愿, 需要对整个产品负责, 在本项目中由组长张津豪担任此角色。
- (2) **Scrum Master (流程管理员)**: 主要负责整个 Scrum 流程在项目中的顺利进行, 同时清除客户和开发工作之间的沟通障碍。需要有一定的技术开发功底, 对开发工作量进行评估, 有一定的编码基础, 另外具有较强的沟通理解能力和组织能力。
- (3) **Scrum Team (开发团队)**: 主要负责在 Scrum 规定流程下进行开发工作, 人数控制在 4~10 人, 每个成员负责不同的技术方面 (包括美工、编码、开发等), 并要求每个成员有很强的自我管理能力和表达能力, 在本项目中由其他组员担任此角色。

4.6.2 开发流程

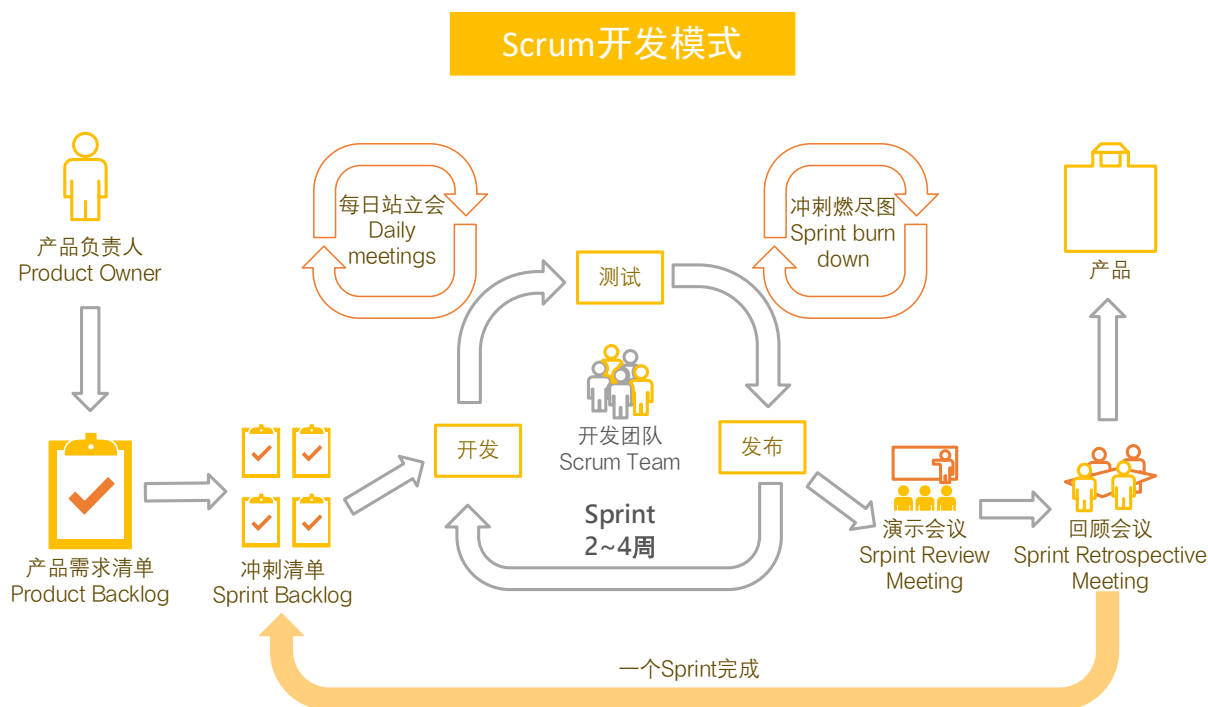


图 13 Scrum 开发流程图

4.6.3 特征

Scrum 的英文意思是橄榄球运动的一个专业术语，表示“争球”的动作。这是一个很形象的比喻，大家工作起来时像打橄榄球一样迅速、富有战斗激情和团队协作。

在整个项目的开发中包含着很多个“冲刺”（Sprint）的过程，在每一次冲刺（15~30天）开发团队要实现一个“冲刺清单”（Sprint Backlog）即创建出软件的一个可用增量。在冲刺的过程中，没有人能够变更“冲刺清单”，这就意味着在一个冲刺中需求是被冻结的，这样使得开发团队的工作更具有目标性。

另外，Scrum 模型对项目的整个实施过程没有过于严格的定义和约束，这就要求开发团队的自我监督。但这样的特性减少了前期项目规划的工作量，使项目开发过程适应性更强、更加灵活。

Scrum 的四条宣言更是精炼的总结了它的特性：^[3]

- （1）个体与交互胜过过程与工具；
- （2）可以工作的软件胜过面面俱到的文档；
- （3）客户协作胜过合同谈判；
- （4）响应变化胜过遵循计划。

4.6.4 优势

- （1）具有开发的高适应性，以人为本，重视团队合作；
- （2）容易学习，应用 Scrum 流程不需要太多的前期投入；
- （3）更加的灵活、更加充分的利用了每个开发者的优势，激发每个人的工作热情。

4.6.5 缺点

- (1) 不重视项目文档，人员出现变动时，交接项目比较困难；
- (2) 没有严格的流程约束，对成员的自我管理能力要求较高；
- (3) 面对面会议与交流较多，对成员的表达能力要求较高。

4.6.6 适用性

首先 Scrum 的开发模式很容易学习，并且不需要前期的项目投入，网络上有免费成熟的 Worktile Agile、Leangoo 等在线协作平台，这对于我们这种小型的开发预算不高的项目来说是十分适用的。

另外，Scrum 的开发模式比较灵活，很适合我们非专业团队的业余开发。较频繁的总结会议，更能监督每个成员的工作进度，并进行思维碰撞。

我们团队中每个成员的优势不同，有的擅长设计美工，有的擅长代码编写，有的擅长组织整合，而 Scrum 的开发模式可以更加充分的利用每个成员的优势，激发大家的热情。

五、软件过程模型选型结论

5.1 选型结论：RAD 模型

5.2 原因阐述

5.2.1 项目开发特点

- (1) 基于学校教务系统的功能，使用学校数据库；
- (2) 账户与用户（学生）一对一；
- (3) 通过模块化的形式使用户能通过点击接口按钮跳转相应板块；
- (4) 项目功能较简单且互相独立（查询、导出），非大型项目；
- (5) 开发人员较少，投入资金不多；
- (6) 项目所需的从设计到运行的时间较短。

5.2.2 RAD 模型优势

- (1) 设计与实践间隔短，一旦确定好需求，实现较容易，对不复杂的内容有效；
- (2) 可以测试原型，有足够用户提供反馈在此基础上进行修改；
- (3) 可复用，可以不断修改完善；
- (4) 适用于对于此类模块鲜明、性能需求不高的项目；
- (5) 项目能够在短时间之内实现（60-90 天）。

六、参考文献

- [1]. 张巍威. Scrum 软件开发方法在 ROSS 公司的应用研究[D]. 华东理工大学, 2014..
- [2]. 王一舒, 蒋冬清, 李三雁. 基于 Scrum 框架的应用型大学实训项目管理初探[J]. 科教导刊(中旬刊), 2016(1).
- [3]. Cat Qi, 《敏捷开发之 Scrum》, <http://qixuejia.cnblogs.com/>
- [4]. 廖昕, 陈松乔, 孙莹. 可复用构件组装技术研究[J]. 计算技术与自动化, 2004(03):51-53.
- [5]. 梅宏, 陈锋, 冯耀东, et al. ABC:基于体系结构、面向构件的软件开发方法[J]. 软件学报, 2003(04):11-22.