计算机学院软件工程与实践课程实验报告

|  |  |  |
| --- | --- | --- |
| 实验题目：持续沟通修订可行性分析报告、CASE调研、软件生命周期模型 | | 学号：201920130238 |
| 日期：2022.3.13 | 班级： 计科19级1班 | 姓名： 陈文盼 |
| Email：2220533881@qq.com | | |
| 实验目的：  1. 持续项目沟通、调查，修订可行性分析报告  2. 调研CASE软件工具，并学会使用  3. 研讨传统软件开发过程模型与敏捷开发 | | |
| 实验软件和硬件环境：  Windows10操作系统 | | |
| 实验步骤与内容：  团队名称：没有bug软工实践团队  团队成员（排名不分先后）：陈文盼 201920130238，史子涵 201900130126，刘诗婷201918130217， 赵一帆 201905130197，郑晓旭 201900130107。   1. 继续项目沟通、调查、对项目内容的补充和完整，完善可行性分析报告。   在上一周，我们已经完成了团队的建立以及与需求方的初步对接，大致确定了客户需求为制作一个基于B/S架构的学生选课管理系统。在本周，我们进一步提出项目内容补充和调整。经过讨论，我们初步掌握了软件的功能：  (1) 要求系统可以准确地录入、删除和查询学生、讲师信息，包括姓名、学号、专业等。  (2) 要求系统可以准确地录入、删除和查询每个讲座的详细信息，包括时间、地点、讲课教师、课容量等。  (3) 要求系统可以查看选择课程的学生信息，并且提供导出学生信息表格的能力， 方便进行签到。  (4) 要求系统可以以表格的形式进行批量的课程导入和学生的信息导入。  (5) 要求系统可以实时修改扩大讲座的课容量。  在接下来的工作中，我们将会不断测试和更新我们的需求和分析。   1. 调研书上和现实中IT公司所采用的的开发过程生命周期模型，讨论自己项目接近应用哪种模型。   在2022年3月17日周四19时，我们小组举行会议，初步分配了四个任务：   1. 分工调研开发过程生命周期模型; 2. 分工撰写针对项目的软件开发计划; 3. 分工调研CASE并选择合适的工具完善可行性分析报告; 4. 讨论项目中可能存在的风险、细化风险。   经过讨论，陈文盼同学负责调研W模型和喷泉模型，cace工具的图表绘制工具；史子涵同学负责调研敏捷方法和可转换模型，case工具的查看源码工具；郑晓旭同学负责调研瀑布模型和原型化模型，case工具的配置管理工具；赵一帆同学负责调研增量和迭代以及螺旋模型，case工具的UML建模工具；刘诗婷同学负责调研V模型和可操作规格说明，case工具的开源分布式版本控制系统。每个人需要理解自己所调研的模型的大致原理和过程，并撰写文稿说明该模型的模型简介、优点和缺点，此外每个人根据选课系统存在的服务器安全风险、网络安全风险、管理安全风险提出一点策略。  在2020年3月20日星期日20时，我们小组通过腾讯会议进行了会议，集中讨论自己的生命周期模型的调研情况、case工具、对项目中存在的安全性风险提出解决策略，具体的调研结果在本篇报告的问题讨论中有所体现。结合调研结果以及我们小组全体对我们所需制作的软件系统的分析，我们最终决定采用V模型作为我们软件开发过程的生命周期模型。   1. 参考软件设计文档国家标准和PPT，补充和完善之前的可行性分析报告。   在第一周实验课上，我们粗略做了可行性分析报告，这周我们更加完善具体的补充了可行性分析报告。每个组员负责一部分，然后通过金山文档汇总。   1. 继续熟悉网络协作平台。将小组工作文档、进度等文档都保存到共享平台上。 | | |
| 结论分析与体会：  本周的实验重点在于对各种不同的软件开发过程生命周期模型的理解，找到合适的CASE工具，以及分析项目中的风险提出解决策略。  为了使规模大、结构复杂和管理复杂的软件开发变的容易控制和管理，人们把整个软件生命周期划分为若干阶段，使得每个阶段有明确的任务，整理出软件生命周期模型。  在我看来，生命周期模型从环节来看可以分成两种：瀑布型生命周期和敏捷类生命周期。对于瀑布型生命周期，在开发过程中只有完成上一阶段的任务后才能进行下一阶段的任务；对于敏捷类生命周期，开发过程中相邻几个阶段可以同时或者迭代进行开发，弥补了传统开发方式中的一些弱点，具有更高的成功率和生产率。从着重点来看可以分成两种：侧重于过程以及侧重于客户交流。前者注重开发过程的完整性，后者以客户的反馈作为下一步任务的出发点和落脚点。随着经济与科技的发展，越来越多不同需求、不同模型的软件需求出现，相对应的不同类型的开发模型也越来越多。 | | |
| 就实验过程中遇到的问题及解决处理方法，自拟1－3道问答题：  一．各种常见的软件开发过程生命周期模型简介和它们的优缺点？  V模型  V模型是瀑布模型的变种，他说明测试活动是如何与分析和设计相联系的。如下图所示，编码位于V型符号的顶点，分析与设计在左边，测试与维护在右边。该模型中连接V型符号左边和右边的连线意味着，如果在验证和确认期间发现了问题，那么在再次执行右边的测试步骤之前，重新执行左边的步骤以修正和改进需求、设计和编码。换言之，V模型 使得隐藏在瀑布模型中的迭代和重做活动更加明确，瀑布模型关注的通常是文档和制品，而V模型关注的是活动和正确性。    优点：  1.使得隐藏在瀑布模型中的迭代和重做活动更加明确。  2.说明了测试活动是如何与分析和设计相联系的。  缺点：  1.仅仅把测试过程作为在需求分析、系统设计及编码之后的一个阶段。  2.忽视了测试对需求分析,系统设计的验证，一直到后期的验收测试才被发现。  可操作规格说明  可操作规格说明模型通过演示系统行为的方式来评估或执行系统需求。也就是说，一旦指定了需求，就可以用软件包进行演示。这样，在设计开始之前就可以评价它们的隐含含义。该模型的过程与诸如瀑布模型这样的传统模型有很大的不同。瀑布模型把系统的功能与设计分离，目的是把客户的需求与实现分开，而可操作规格说明模型允许把功能和设计合并起来。该模型图示如下。    优点：  1.允许用户和开发人员在早期检查需求。  2.确定需求后，可通过软件包进行演示。  缺点：  1.开发周期较长，修改次数多。  可转换模型  定义：通过去除某些主要开发步骤来设法减少出错的机会。利用自动化手段的支持，转换过程使用一系列转换把需求规格说明变为一个可交付使用的系统。其图形化表示为：    缺点：需要一个精确表述的形式化规格说明，这样才可以基于它进行操作。  优点：见定义。  敏捷方法  提出原因：许多软件开发方法都试图在软件构思、文档化、开发和测试的过程中强加某种形式的严格性，为了强调灵活性在快速有效的软件生产中所发挥的作用。  四条原则：  1.个体和交互的价值胜过过程和工具；  2.可以工作的软件胜过面面俱到的文档；  3.客户合作胜过合同谈判；  4.响应变化胜过合同谈判；  典型方法：极限编程、水晶法、并列争球法、自适应软件开发。  极限编程（XP）：  四个特性：交流、简单性、勇气和反馈  十二条准则：规划游戏，小的发布，隐喻，简单设计，首先编写测试，重构，对编程，集体所有权，持续集成，可以忍受的步伐，在现场的客户，代码标准。  缺点：其有效性的证据很少，很多极限编程的实践是相互依赖的，如果其中一个被修改，其他的都会受到影响。  并列争球法（Scrum）  定义：一种迭代的循序渐进的开发方法。以需求为中心，注重交互和协作，积极响应需求的变化。按照需求的级别来实现产品。通过频繁交付和过程控制来提高产品的竞争优势。减少文档，努力实现全局视图和软件源代码一起演化。  与瀑布模型对比的优势：  1.灵活性强：开发方法强调灵活，主动拥抱需求变化，能够迅速反应用户需求；  2.开发成本低：降低了文档维护成本，交流沟通成本和时间成本；  3.最大化生产率：以有价值的交付为核心目标，能迅速反应市场需求；  4.项目风险低：开发交付时间短，产品迭代速度快，可以有效应对市场变化，降低项目风险；  螺旋模型：  主要特点：  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。    图中的螺旋线代表随着时间推进的工作进展；开发过程具有周期性重复的螺旋线形状。4个象限分别标志每个周期所划分的4 个阶段：制定计划、风险分析、实施工程和客户评估。从制定计划开始，然后风险分析，如果通过了风险分析，则进入开发阶段，开发完成后再进行评估。评估后又进入外一层螺旋进行下一次循环迭代，制定计划，风险分析……直到项目实现。风险分析是其最大特点。  优点：  1、将待开发的软件系统模块化，可以分批次地提交软件产品，使用户可以及时了解软件项目的进展。  2、以组件为单位进行开发降低了软件开发的风险。一个开发周期内的错误不会影响到整个软件系统。  3、开发顺序灵活。开发人员可以对组件的实现顺序进行优先级排序，先完成需求稳定的核心组件。当组件的优先级发生变化时，还能及时地对实现顺序进行调整。  缺点：  1、如果待开发的软件系统很难被模块化，那么将会给增量开发带来很多麻烦。  2、需要分析人员有风险分析的经验，因为分析失误会造成很大的损失。  3、周期相对较长，所以不适用于一般软件开发项目。  增量模型  主要特点：  增量模型是把待开发的软件系统模块化，将每个模块作为一个增量组件，从而分批次地分析、设计、编码和测试这些增量组件。运用增量模型的软件开发过程是递增式的过程。相对于瀑布模型而言，采用增量模型进行开发，开发人员不需要一次性地把整个软件产品提交给用户，而是可以分批次进行提交。增量模型又称为有计划的产品改进模型，它从一组给定的需求开始，通过构造一系列可执行中间版本来实施开发活动。第一个版本纳入一部分需求，下一个版本纳入更多的需求，依此类推，直到系统完成。每个中间版本都要执行必需的过程、活动和任务。  https://bkimg.cdn.bcebos.com/pic/1b4c510fd9f9d72a972b6afadd2a2834359bbb8e?x-bce-process=image/watermark,g_7,image_d2F0ZXIvYmFpa2U5Mg==,xp_5,yp_5  优点：1、由于能够在较短的时间内向用户提交一些有用的工作产品，因此能够解决用户的一些急用功能。  2、由于每次只提交用户部分功能，用户有较充分的时间学习和适应新的产品。  3、对系统的可维护性是一个极大的提高，因为整个系统是由一个个构件集成在一起的，当需求变更时只变更部分部件，而不必影响整个系统。  缺点：1、由于各个构件是逐渐并入已有的软件体系结构中的，所以加入构件必须不破坏已构造好的系统部分，这需要软件具备开放式的体系结构。  2、在开发过程中，需求的变化是不可避免的。增量模型的灵活性可以使其适应这种变化的能力大大优于瀑布模型和快速原型模型，但也很容易退化为边做边改模型，从而使软件过程的控制失去整体性。  3、如果增量包之间存在相交的情况且未很好处理，则必须做全部系统分析。这种模型将功能细化后分别开发的方法，较适应于需求经常改变的软件开发过程，不太适合需求基本不改变的软件。  迭代模型：  主要特点：  迭代模型是RUP（统一软件开发过程）推荐的周期模型。迭代包括产生产品发布的全部开发活动和要使用该发布必需的所有其他外围元素，也就是对产品开发的每个活动、阶段或子阶段进行迭代。整个开发工作被组织为一系列的短小的、固定长度的小项目，被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试。采用这种方法，开发工作可以在需求被完整地确定之前启动，并在一次迭代中完成系统的一部分功能或业务逻辑的开发工作。再通过客户的反馈来细化需求，并开始新一轮的迭代。    优点：1、降低了在一个增量上的开发风险。如果某个迭代完成后的软件不符合客户要求，那么损失只是这一个开发有误的迭代的花费。  2、降低了产品无法按照既定进度进入市场的风险。通过在开发早期就确定风险，可以尽早来解决而不至于在开发后期匆匆忙忙。  3、加快了整个开发工作的进度。因为开发人员清楚问题的焦点所在，他们的工作会更有效率。  4、由于用户的需求并不能在一开始就作出完全的界定，它们通常是在后续阶段中不断细化的。因此，迭代过程这种模式使适应需求的变化会更容易些。  缺点：  1、对产品人员的节奏把控能力（定每周目标，需求优先级剖析，以及临时需求的处理）有较高要求。   1. 适合适合需求不明确、架构风险大的项目，不适合需求明确、架构稳定的项目。   W模型：  简介：测试伴随着整个软件开发周期，而且测试的对象不仅仅是程序，需求、设计等开发输出的文档同样要测试（这里针对设计文档，一般可以划分为需求设计文档、概要设计文档、详细设计文档和代码文档），也就是说，测试与开发是同步进行的。由两个V字型模型组成，分别代表测试与开发过程，图中明确表示出了测试与开发的并行关系。  IMG_256  优点：  1.测试的活动与软件开发同步进行  2.测试的对象不仅仅是程序，还包括需求和设计  3.尽早发现软件缺陷可降低软件开发的成本  缺点：  1.在W模型中，需求、设计、编码等活动被视为串行的，同时，测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一个阶段工作。这样就无法支持迭代的开发模型。  2.对于当前软件开发复杂多变的情况，W模型并不能解除测试管理面临的困惑。  喷泉模型  简介：喷泉模型主要用于采用对象技术的软件开发项目。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。软件的某个部分常常被重复工作多次，相关对象在每次迭代中随之加入渐进的软件成分。无间隙指在各项活动之间无明显边界，如分析和设计活动之间没有明显的界限，由于对象概念的引入，表达分析、设计、实现等活动只用对象类和关系，从而可以较为容易地实现活动的迭代和无间隙，使其开发自然地包括复用。  IMG_256  优点：  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。  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。  瀑布模型  1.定义：  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，那么最好 “返回”上一个阶段并进行适当的修改，项目开发进程从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。包括软件工程开发、企业项目开发、产品生产以及市场销售等构造瀑布模型。  大致过程如下图：  IMG_256  2.特点  2.1 阶段间具有顺序性和依赖性：  必须等前一阶段的工作完成之后，才能开始后一阶段的输入。对本阶段工作进行评审，若得到确认，则继续下阶段工作，否则返回前一阶段，甚至更前阶段。只有前一阶段输出正确，后一阶段才能正确。  2.2 推迟实现的观点：  在编码之前，设置了需求分析与设计的各个阶段，分析与设计阶段的根本任务规定在这两个阶段主要考虑目标系统的逻辑模型，不涉及软件的物理实现。  2.3质量保证的观点：每个阶段都坚持两个做法：  规定文档，没有文档就没有完成该段任务。  每个阶段结束前都要对完成的文档进行评审，以便尽早发现问题，改正错误。  3.缺点  3.1依赖于早期进行的唯一的一次需求调查，不能适应需求的变化；  3.2由于是单一流程，开发中的经验教训不能反馈应用于本产品的过程；  3.3风险往往迟至后期的开发阶段才显露，因而失去及早纠正的机会。  4.适用项目  4.1需求清晰明了且软件开发时间相对宽松的项目  4.2规模小，需求简单，功能单一的项目  5.阶段划分  设计阶段->编码阶段->测试阶段->发布阶段->实施阶段->运营阶段  原型化模型  1.定义  原型模型快速建立起来的可以在计算机上运行的程序，他所能完成的功能往往是最终产品能完成的功能的一个子集。一般来说，根据客户的需要在很短的时间内解决用户最迫切需要，完成一个可以演示的产品，这个产品只实现部分功能。原型最重要的是为了确定用户的真正需求。  原型化模型的阶段图大致如下：  IMG_256  原型化模型常有以下两种形式：  抛弃型：开发原型为了获取需求，在原型开发之后，已获取了更为清晰的需求信息，原型无需保留而废弃；  渐进型：原型作为软件最终产品的一部分，可满足用户的部分需求，进一步在此基础上开发，则可增加需求，实现后再交付使用；  2.特点  2.1用户需求不完全或不确定；  2.2针对总体的轮廓先建立一个用户需求原型，然后进行评价和反馈；  2.3对原型进行扩充、改进和求精；  2.4完成最终系统  3.缺点  3.1没有考虑软件的整体质量和长期的可维护性。  3.2大部分情况是不合适的操作算法被采用目的为了演示功能，不合适的开发工具被采用仅仅为了它的方便，还有不合适的操作系统被选择等等。  3.3由于达不到质量要求产品可能被抛弃，而采用新的模型重新设计。  4.适用项目  4.1客户能提出一般性的目标，但不能标出详细的输入、处理及输出需求；或开发者不能确定算法的有效性、操作系统的适应性、及人机交互的形式。  4.2用户定义了一组一般性目标，但不能标识出详细的输入、处理及输出需求；  4.3开发者可能不能确定算法的有效性、操作系统的适应性或人机交互的形式  5.阶段划分  5.1.抛弃型原型模型的阶段划分：  需求分析阶段----获取业务需求  原型实现阶段—--主要是界面实现，业务流程用图形方式表示。  客户评价阶段----和客户确认，完善业务需求  5.2.渐进型原型模型的阶段划分：  需求分析阶段（需求分析、原型实现、客户评价）->设计阶段->编码阶段->测试阶段->发布阶段->运营阶段。  二．Case 工具：  图表绘制软件：SmartDraw  相比于visio等软件，SmartDraw更适合在校学生使用，向用户提供了大量可用的模板，可以极大程度上方便我们的工作，节省更多时间。此外，我们还可以去该公司的网站下载更多的符号和外形，总量达数百兆之巨，完全足够满足我们项目制作各类图表的需要。使用smart draw，我们可以直接模板中选择所需要的，然后用简单的命令来添加信息，之后SmartDraw会帮我们完成剩下的事。  查看源码工具：Source Insight  因为通过工程的管理，SI可实现多文件代码中的变量、函数的快速定位和搜索，并且对每个打开的源程序（C或C++代码）中的变量和函数的程序语句进行彩色显示等功能，便于我们在一个相对而言很大的工程中快速定位自己所要寻找的函数和程序段、定位变量和函数的定义和声明，非常方便于我们上下文跳转去阅读，可以大大节省我们的时间。  配置管理工具：Visual Source Safe  因为VSS可以记录每个上传的文件的所有改动版本，允许回溯到以前版本的文件，可以防止用户无意中丢失文件，这是我们团队很需要的功能，保证了项目文件的安全性。同时，VSS也允许分支、共享、合并和管理文件版本，具有的功能已经可以满足我们团队的需求。并且依据介绍显示VSS最适合团队的规模较小，各个成员之间的地理距离比较近，通常在高度可靠的环境中通过高速、低延迟的局域网工作情形下，是一种仅用于客户端的文件服务器应用程序，不需要服务器端处理或代码执行。根据描述完全适配于我们的项目开发团队。  UML建模工具：Rational Rose  Rational Rose支持Java，J2EE，C++，MCF等语言和框架的建模。Rose 提供了与所有主导的 IDE、编程语言、数据库系统和配套技术的无缝集成。一旦熟悉了 Rational Rose，我们就可以轻松在项目间转换，更高效迅速地完成任务。  开源分布式版本控制系统:git  队内其他开发人员往Github项目中修改了代码，若我们需要在他们的基础上继续进行开发，使用git把Github的代码同步下来，同时使用git将修改后的内容push上去，实现对GitHub仓库中的代码进行更新。  附：  git init 文件夹进行git初始化，必须要有这一步才可以进行git操作（因为没有这一步就没有.git文件夹）  git pull xxxxx.com master 拉去xxxxx网站项目的master分支  git add . 将本地的项目保存到git文件夹里  git commit -m "xxxx" 进行一次提交的操作（并没有真正提交上去）  git push xxxx.com master 将commit的文件提交到master分支  上述提交方法是不规范的，因为这么操作完之后，修改后的内容直接覆盖了master分支，对于合作开发而言是不友好的。  在pull分支之后进行下面的操作  git branch mybranch 创建一个名字叫做mybranch分支名的分支  git checkout mybranch 切换到mybranch分支然后进行修改，注意这个时候我们操作的分支是mybranch而不是master分支 所以在提交的时候需要这样：  git push xxx.com mybranch  三．可能存在的风险及其风险管理  1.开发人员因病或者其他事务而不能参与开发工作。本次工作人员储备很充足，如果在事前缺席开发工作，仍然能够很轻松的找到代替人员。  2.开发人员因病或者其他事物中途退出整个开发工作。本次开发工作的工作人员建设上，实际上有一定的松弛度，当有一个人缺席的情况下，其余四人仍然能够在压力范围之内完成任务。也不必担心其他人不熟悉缺席人的工作内容而无法接手。每天工作之后，组员都会讨论各自任务部分的建设，所以大家都对整个项目的各个部分有了一定得了解。能够在缺失人手时顶住压力。整个团队的抗压能力比较大。  3.设备损坏。这个是最好处理的问题，当下有很多的备用设备以供选择。  4.力不足无法按时完成开发任务。这个是不必担心的。在接手这次开发任务之前，就已经对这个项目进行了可行性分析，确认是我们团队能够解决的才会去投标。  5.无法满足委托方要求。在进行产品的功能分析的时候，就已经与委托方进行过确认，所有功能都是按照委托方的要求来的，如果委托方不满意，属于委托方违约。并且在测试阶段，也会根据委托方的合理要求进行已成程度上的修改，直至委托方满意。如果委托方还是不满意，那就只能按照合约加时或者提出委托方违约。  6. 系统性能风险。在进行项目开发之前先设计和搭建出系统的基础架构并进行性能测试，确保架构符合性能指标后再进行后续工作。  7. 项目组成员之间发生冲突，导致沟通不畅、设计欠佳、接口出现错误和额外的重复工作。项目在建设之初就将项目目标、工作任务等和项目成员沟通清楚，采用公平、公正、公开的绩效考评制度，倡导团结互助的工作风尚等。 | | |