**实验四** **XP开发方法，DevOps，活动图**

202100031020 夏思晓

**一、实验目的：**

1.了解XP开发方法

2. 了解DevOps

3. 理解项目活动图

**二、实验内容：**

1. 阅读XP开发方法文档，理解XP过程工作模型。

**（1）XP概述**

极限编程（Extreme Programming，简称XP）是敏捷软件开发方法的代表，是一种轻量级的、灵巧的、简单的软件工程方法。与传统的开发过程不同，极限编程的核心活动体现在需求→测试→编码→设计过程中。因此适用于规模小、进度紧、需求变化大、质量要求严的项目。它希望以最高的效率和质量来解决用户目前的问题，以最大的灵活性和最小的代价来满足用户未来的需求。

XP将一个软件开发项目视为一个包含四个控制“变量”的系统：成本、时间、质量和范围，变量之间的关系如图1所示。

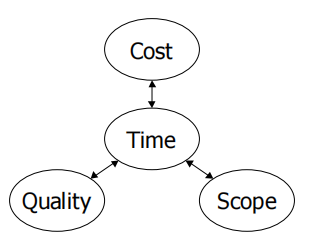


图 1 变量之间的关系

极限编程是价值而非实践驱动的高度迭代的开发过程。其价值体现在以下几个方面：第一，**沟通（Communication）**：即追求有效的沟通。XP强调项目开发人员、设计人员、客户之间等有效地、及时地沟通，确保各种信息的畅通。第二，**简单（Simplicity）**：即实现最简单的可行方案。XP认为应该尽量保持代码的简单，只要能够满足工作需要就行，这样有利于代码的重构和优化。第三，**反馈（Feedback）**：即快速有效的反馈。要求不断对当前系统状态进行反馈，通过反馈，达到迅速沟通、编码、测试、发布的目的。第四，**勇气（Courage）**：即勇于放弃和重构。对于用户的反馈，XP程序员要勇于对自己的代码进行修改，即使有些修改可能会使得原来已经通过的测试又出现错误，但是经过团队的共同攻关，最终必然会取得满意的效果。

1. **XP的开发过程及特点**

与传统的开发过程不同，极限编程的核心活动体现在需求→测试→编码→设计过程中，因此对工作环境、需求分析、设计、编程、测试、发布等提出了新的思路和要求。

**①工作环境**：XP要求每个参加项目开发的人都担任一个角色（项目经理、项目监督人等），并履行相应的权利和义务。所有的人都在一个开放式的开发环境中工作，最好是在同一个大房间中工作，随时讨论问题，强调每周40小时工作制，不加班。

**②需求分析**：客户被纳入开发队伍。由于客户不具备计算机专业知识，无法用专业语言明确描述需求，所以开发人员和客户一起，用讲故事的方式把需求表达出来，这种故事被称为user story，即用user story表示需求。开发人员根据经验将许多user story 组合起来，或将其进行分解，最终记录在story card的小卡片上，这些user story将陆续被程序员在各个小的周期内，按照商业价值、开发风险的优先顺序逐个开发。

**③设计**：XP强调简单设计（simple design），即用最简单的办法实现每个小需求。在XP中，没有那种传统开发模式中一次性的、针对所有需求的总体设计，这些设计只要能够满足系统客户在当前的需求就可以了，不需要考虑将来可能的变化，整个设计过程包括在整个螺旋式发展的项目中。

**④编程**：成对编程（pair programming）是极限编程的一大特色，即两个人一起使用同一个屏幕，同一个键盘，共同完成一段程序的编码。成对编程的好处是，可以提高纪律性，更容易写出优质的代码，同时保证编程的流畅进行，更重要的是，能够使得整个团队更方便地分享编程经验，有利于新手的快速成长。

**⑤测试**：在极限编程中，测试是非常重要的一个环节，它首先要求在开始写程序之前先写好测试，其目的是为了提高软件的可测试性。XP要求开发人员经常把开发好的模块整合到一起，每次整合后都要运行单元测试；做任何的代码复核和修改，都要运行单元测试；发现了漏洞，就要增加相应的测试。除了单元测试之外，还要进行整合测试、功能测试、负荷测试和系统测试等。所有这些测试是极限编程开发过程中最重要的文档之一，也是最终交付给用户的内容之一。

**⑥发布**：XP要求按照开发计划，每经过一个开发周期，软件就发布一次，而不是像传统的开发方法那样，整个软件开发完成后才发布。在一个开发周期内，开发人员要求客户选择最有价值的user story作为未来一两个星期的开发内容，一个开发周期完成后，提交给客户的系统虽然不是最终的产品，但它已经实现了几个客户认为是最重要的story，开发人员将逐步在其基础上增加新的模块，而且在发布前软件都经过单元测试和集成测试，因此，虽然软件并不完备，但是，发布的软件客户还是可以真正使用的。

1. **XP的生命周期**

XP的生命周期支持并鼓励持续集成，这意味着团队成员会非常频繁地进行集成，可能每小时或每天都会进行一次。总结来看，XP生命周期的完整流程包括以下步骤：

* 从用户故事中提取尚未完成的工作；
* 将重点放在最重要的事项上；
* 开始进行迭代规划；
* 实施真诚和实事求是的规划；
* 与所有相关方保持持续的沟通，并授权团队成员；
* 发布或交付工作成果；
* 收集并接受反馈；
* 根据收到的反馈返回到迭代规划阶段，并根据需要重复整个过程。

1. **XP的五条规则**

**①规划。**在极限编程（XP）的规划阶段，团队必须评估项目是否适合采用XP方法。这包括：

* 审查用户故事以确保它们简单并且客户能够参与整个开发过程。如果用户故事太复杂或是由不参与的客户提供的，那么XP可能不是合适的方法。
* 确认项目的商业价值和优先级，以保证团队首先完成最重要的工作。
* 考虑项目所处的开发阶段。XP通常更适合项目的早期开发阶段，而不是后期的迭代阶段。

一旦确定项目适合采用XP，团队应该创建一个发布计划。在这个过程中，团队应该：

* 将项目分解成多个迭代，并为每个迭代制定详细计划。
* 设定现实的截止日期，并确保工作步调是可持续的。
* 随着项目的进展，及时分享更新，这样做可以增强团队的诚实和透明度。
* 分享实时的进度更新，以帮助团队识别问题、适应变化并更快地做出响应。
* 使用项目管理工具，如看板或时间线，来实时跟踪进度。

**②管理。**在极限编程（XP）中，管理团队的物理空间是一个关键因素。XP的理想情况是所有团队成员在一个开放的工作空间中共同工作，以促进协作。然而，在远程工作的环境中，团队可能需要使用支持异步工作的远程协作平台。

与其他敏捷方法相似，XP推荐使用**每日站立会议**来促进持续和开放的沟通。团队应该实施每周和每季度的循环。在季度循环中，团队会回顾指导工作的用户故事，并审视XP过程中的差距或改进机会。每周循环以客户会议开始，客户在会议中选择他们希望团队在那一周专注的用户故事。

作为经理或团队领导，需要专注于维护工作进度和测量步调，根据出现的挑战或问题调整团队成员的分配，或者根据当前项目和迭代的需要调整XP过程。XP的目标是保持灵活性并迅速采取行动，所以作为领导者，需要密切关注团队的当前工作并对任何变化做出响应。

**③设计。**在开始极限编程（XP）的设计阶段时，应该从最简单可能的设计开始，并认识到后续的迭代将会使设计变得更加复杂。在这个阶段，应避免添加不必要的早期功能，以保持设计的简单性。

在使用XP方法时，团队通常会使用类-职责-协作（CRC）卡片来帮助展示设计中每个对象的作用和相互作用。CRC卡片的填写过程可以帮助团队成员可视化所有功能的交互，因为这些功能是相互关联和作用的。CRC卡片通常包含以下内容：

* 类 – 表示相似对象的集合。
* 职责 – 列出与类相关的职责。
* 协作者 – 指出与该类交互的其他类。

CRC卡片是一个有用的工具，可以帮助激发设计过程中的创意并发现潜在的问题。在设计过程中，你应该使用一个系统来减少潜在的瓶颈。为了做到这一点，你需要积极地寻找和识别风险。一旦识别出潜在的威胁，应该指派一到两个团队成员来找到解决方案，以防这些威胁变成实际问题。

**④编码。**在极限编程（XP）中，编码阶段的一个显著特点是开发者需要与客户保持持续的沟通。这种合作方式允许开发者在每次迭代中即时地测试和整合客户的反馈，而不是等到一个开发周期结束。XP中的编码规则是严格的，包括以下几点：

* 所有编写的代码都必须遵守既定的编码标准。
* 使用单元测试来验证需求，并在项目的所有方面进行开发。
* 采用成对编程的方法，即两名开发者共同在一台电脑上工作，这样做不会增加开发时间，反而通过双倍的专注力来确保代码质量。
* 实施持续集成，以便新代码可以被立即添加并测试。
* 在任何给定时间，只有一对开发者可以更新代码，这样做可以减少错误的发生。
* 实行集体代码所有权，即团队中的任何成员都可以随时更改任何人的代码。

**⑤测试。**在XP的整个过程中，测试是不断进行的。所有的代码在发布之前都需要通过单元测试。如果在单元测试中发现错误，就会创建新的测试来修复这些错误。之后，开发者会将他们一直在处理的用户故事配置进验收测试中。在验收测试中，客户会审查结果，以确认开发者如何将用户故事转化为实际的产品功能。

1. **XP的十三个最佳实践**

在XP中，集成了13个最佳实践，它们没有一个是创新的概念，但其主要创新点在于提供一种良好的思路，将这些最佳实践结合在一起，并且确保尽可能彻底地执行它们，使得它们能够在最大程度上相互支持。13个最佳实践主要包括：计划游戏、小型发布、隐喻、简单设计、测试先行/测试驱动开发、重构、结对编程、集体代码所有制、持续集成、每周工作40小时/可持续的速度、现场客户、编码标准、配合是关键。

1. **XP的优缺点**

**①极限编程的优点：**与传统的软件工程方法相比较，极限编程重视客户的参与；重视团队合作和沟通；制定计划前做出合理预测；让编程人员参与软件功能的管理；重视质量；简单设计；高频率的重新设计和重构；高频率及全面的测试；递增开发；连续的过程评估；对过去的工作持续不断的检查。

**②极限编程的缺点：**以代码为中心，忽略了设计；缺乏设计文档，局限于小规模项目；对已完成工作的检查步骤缺乏清晰的结构；质量保证依赖于测试；缺乏质量规划；没有提供数据的收集和使用的指导；开发过程不详细；全新的管理手法带来的认同度问题；缺乏过渡时的必要支持。

1. **XP的应用**

XP适用于规模小、进度紧、需求变化大、质量要求严的项目。它希望以最高的效率和质量来解决用户目前的问题，以最大的灵活性和最小的代价来满足用户未来的需求，XP在平衡短期和长期之间做了巧妙的安排。

2. 阅读DevOps文档，了解DevOps

1. **DevOps概述**

DevOps一词的来自于Development和Operations的组合，DevOps是为了填补开发端和运维端之间的信息鸿沟，改善团队之间的协作关系，通过自动化流程来使得软件构建、测试、发布更加快捷、频繁和可靠。

首先，DevOps 是一种文化变革，旨在打破开发和运维之间的隔阂，强调团队之间的协作和沟通。DevOps 的核心理念是持续集成、持续交付和持续部署，通过自动化的方式加快产品的迭代速度，提高产品质量和用户满意度。DevOps 还强调自动化测试和监控，以便及时发现和解决问题，提高系统的可靠性和稳定性。

其次，DevOps 是一种敏捷开发方法，强调快速响应变化和灵活性。DevOps 鼓励快速迭代、小步快跑，并通过自动化的方式实现测试、部署和运维。DevOps 可以帮助团队更好地适应市场变化，快速响应用户需求，提高产品的竞争力。

最后，DevOps 是一种技术实践，涉及多个工具和平台的使用，如容器化技术、微服务架构、CI/CD 管道等。DevOps 可以帮助团队更好地管理和维护系统，提高系统的可扩展性和可维护性。

总之，DevOps 是一种文化、方法和技术的集合体，旨在提高软件交付的速度和质量，增强团队协作和沟通，提高系统的可靠性和稳定性。DevOps 的核心是持续改进，不断优化流程、工具和实践，以适应不断变化的业务需求和技术环境。

1. **DevOps的五个关键原则**

**①协作。**DevOps 背后的关键前提是协作。开发和运营团队合并为一个职能团队，在整个开发和部署周期中进行沟通、共享反馈和开展协作。通常，这意味着开发和运营团队合并成一个团队，并在整个应用生命周期中开展工作。

DevOps 团队的成员负责确保在产品的各个方面交付高质量的成果。这会产生更多的“全栈”开发，即团队承担一项功能或产品的全部后端至前端的责任。从构思到交付的整个生命周期，团队对一项功能或项目全面负责。这加强了团队的投入力度和依附度，进而可提高输出的质量。

**②自动化。**DevOps 的其中一项基本实践是：尽可能实现软件开发生命周期的自动化，让开发人员有更多时间来编写代码和开发新功能。自动化是 [CI/CD 管道](https://www.atlassian.com/zh/continuous-delivery/principles/pipeline)的一个关键要素，它有助于减少人为错误并提高团队工作效率。借助自动化流程，团队可以在较短的迭代时间内实现持续改进，从而快速响应客户的反馈。

**③持续管理。**持续改进被确立为敏捷开发实践以及精益制造和改进 Kata 的主要内容。其重点在于注重实验，最大限度减少浪费，同时优化速度、成本和交付便利性。持续改进还与持续交付息息相关，使 DevOps 团队能够不断推送更新，提高软件系统的效率。持续不断地发布新版本，意味着团队会不断推送代码变更，从而消除浪费、提高开发效率并创造更多客户价值。

**④行动以客户为本。**DevOps 团队利用与客户和最终用户的短程反馈回路，围绕用户需求开发产品和服务。DevOps 实践通过使用实时现场监控和快速部署，实现对用户反馈的快速收集和响应。团队可以即刻了解用户与软件系统的实时交互情况，并利用该洞察信息制定进一步的改进措施。

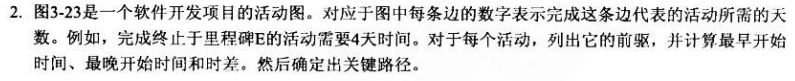
**⑤胸有成竹地进行创造。**此原则在于了解客户需求，创造能解决实际问题的产品或服务。团队不应“在泡沫中构建”，也不应基于对消费者使用软件方式的臆测来开发软件。相反，DevOps 团队应从创建到实施全方位了解产品的各个方面。

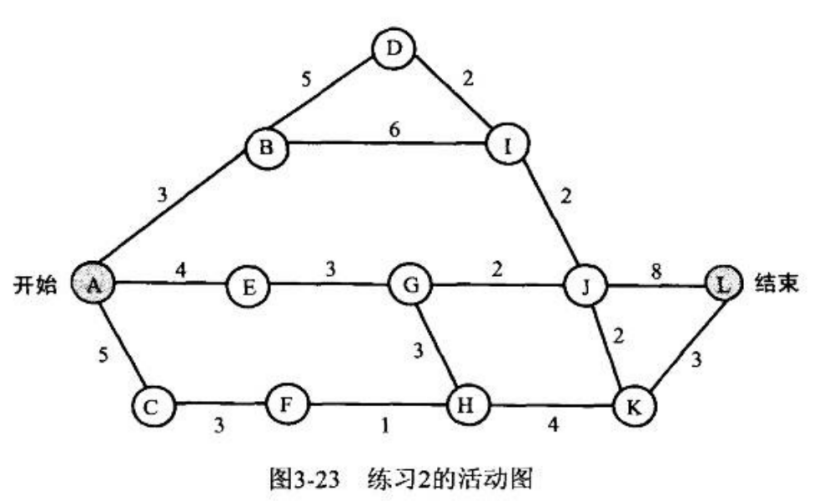
1. **DevOps的优势**

DevOps 的优势主要分为三类：技术优势、文化优势和业务优势。技术优势包括降低复杂性、持续交付和更快解决问题。与传统的孤立团队相比，DevOps 从业人员通常能够更快速地部署更高品质的代码。文化优势是指团队的工作效率更高，客户更加满意。而业务方面的优势则包括团队成员之间加强协作和信任，实现更快的交付速度和稳定的运营环境。

3. 活动图练习

书上练习题2,3（p97-98) 的软件开发项目活动图，找出关键路径。



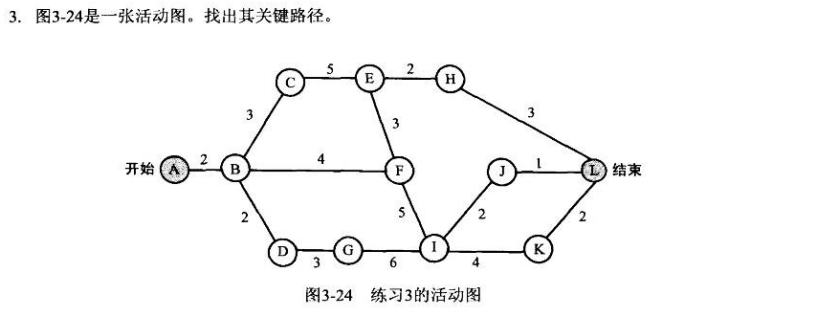


|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 事件 | A | B | C | D | E | F | G | H | I | J | K | L |
| 最早开始时间 | 0 | 3 | 5 | 8 | 4 | 8 | 7 | 10 | 10 | 12 | 14 | 20 |
| 最晚开始时间 | 0 | 3 | 9 | 8 | 7 | 12 | 10 | 13 | 10 | 12 | 17 | 20 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 活动 | AB | AC | AE | BD | BI | CF | DI | EG | FH | GH | GJ | HK | IJ | JK | JL | KL |
| 前驱 |  |  |  | AB | AB | AC | AB,BD | AE | AC,CF | AE,EG | AE,EG | AC,AE,CF,EG,FH,GH | AB,BD,  BI,DI | AB,BD,BI,DI,IJ,AE,EG,GJ | AB,BD,BI,DI,IJ,AE,EG,GJ | AB,BD,BI,DI,IJ,AE,EG,GJ,JK,  AC,CF,FH,GH,HK |
| 最早开始时间 | 0 | 0 | 0 | 3 | 3 | 5 | 8 | 4 | 8 | 7 | 7 | 10 | 10 | 12 | 12 | 14 |
| 最晚开始时间 | 0 | 4 | 3 | 3 | 4 | 9 | 8 | 7 | 12 | 10 | 10 | 13 | 10 | 15 | 12 | 17 |
| 时差 | 0 | 4 | 3 | 0 | 1 | 4 | 0 | 3 | 4 | 3 | 3 | 3 | 0 | 3 | 0 | 3 |
| 关键活动 | \* |  |  | \* |  |  | \* |  |  |  |  |  | \* |  | \* |  |

关键路径：AB→BD→DI→IJ→JL

总长度：20



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 事件 | A | B | C | D | E | F | G | H | I | J | K | L |
| 最早开始时间 | 0 | 2 | 5 | 4 | 10 | 13 | 7 | 12 | 18 | 20 | 22 | 24 |
| 最晚开始时间 | 0 | 2 | 5 | 9 | 10 | 13 | 12 | 21 | 18 | 23 | 22 | 24 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 活动 | AB | BC | BD | BF | CE | DG | EF | EH | FI | GI | HL | IJ | IK | JL | KL |
| 最早开始时间 | 0 | 2 | 2 | 2 | 5 | 4 | 10 | 10 | 13 | 7 | 12 | 18 | 18 | 20 | 22 |
| 最晚开始时间 | 0 | 2 | 7 | 9 | 5 | 9 | 10 | 19 | 13 | 12 | 21 | 21 | 18 | 23 | 22 |
| 关键活动 | \* | \* |  |  | \* |  | \* |  | \* |  |  |  | \* |  | \* |

关键路径：AB→BC→CE→EF→FI→IK→KL

总长度：24

小组讨论，针对自己项目中的工作进行工作活动分解，分工进行各自合理的工作进度估算，最后汇总绘出项目活动图，找出关键路径。（见项目活动图汇总）

将小组工作文档、进度等文档保存到每个小组选定的协作开发平台上。