**实验四** **XP开发方法，DevOps，活动图**

**陈煜202100031018**

**一、实验目的**

**1. 了解****XP开发方法**

**2. 了解DevOps**

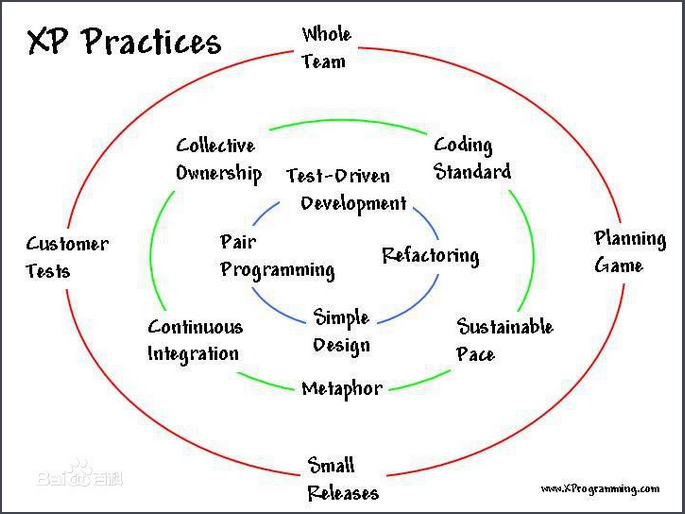
**3. 理解项目活动图**

**二、实验内容：**

**1. 阅读XP开发方法文档，理解XP过程工作模型**

Extreme Programming（极限编程，简称XP）是由KentBeck在1996年基于增量模型发展而提出的。是一种近螺旋式的开发方法，将复杂的开发过程分解为一个个相对比较简单的小周期，将系统细分为多个可以在较短周期解决的子模块，且强调测试、代码质量和及早的发现问题。通过积极的交流、反馈以及其它一系列的方法，开发人员和客户可以非常清楚开发进度、变化、待解决的问题和潜在的困难等，并根据实际情况及时地调整开发过程。

所谓极限，就是所有环节都做到极致，花最短的时间，用最简单的方法做最好的软件。查询大量资料后，XP让我感觉它让软件开发变得轻松愉快并富有有活力。从它“沟通、简单、反馈、勇气和谦逊”的核心价值就可以看出。并且XP是基于敏捷开发的核心价值和目标的，而敏捷开发是以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发的。基于这样的思想，XP也对开发团队作了13个核心实践的要求（如图所示，图片来源百度百科）。



首先是团队合作。每种软件开发过程模型都强调团队合作，但XP与众多模型不同的是，只要是对项目作出贡献的人，都是团队中的一员，其中包括用户。因为从项目的计划到最后验收，用户才是最清楚自己需求的人。并且不是每个成员的工作别人不能插手，而是互相交流协作的。

其次是结对编程。在XP中，所有的代码都是由两个程序员在同一台机器上一起写的。或许这一点很难理解，虽然我们常说人多力量大。但是在开发软件项目这件事情上，我们从来都不希望和别人一起编写同一段代码。因为每个人的想法和编程习惯都不一样。但使用XP模型确强行要求执行这一点，而后来事实也证明，这样极大地提高了工作强度和工作效率。毕竟这样做保证了所有的代码、设计和单元测试都至少被另一个人检查复核过。并且在项目开发过程中，对人员流动较大的项目，这是一条很好的策略。

最后是集体拥有代码。在许多模型中，项目开发之初都会对开发人员进行功能模块的分工，对应功能模块的开发人员只负责维护自己功能，并且他们也不喜欢别人随意修改自己的代码。这就导致开发团队人员互相之间都不了解彼此的功能，也不熟悉彼此的代码。这就使得代码维护具有很大的局限性（因为可能由于负责某一功能的程序员技术的局限性导致维护困难）。但XP中却是大家共同拥有代码，在前面也已经提到，每个成员都可以阅读别人的代码，并发现和纠正其中的错误。这样所有代码都是整个团队共同开发完成的，而不是只是一两个技术较牛的人写的。并且由整个团队所有人开发出来的系统，代码质量都非常好。当然，这样做的基础是每个成员都严格遵循项目开发的代码规范。

极限编程的过程包括计划项目、验收测试、小规模发布。

计划项目（PlanningGame）：主要预测交付日期前可以完成多少工作，现在和下一步的工作内容有哪些。针对这两个问题，XP中又提出了软件发布计划和周期开发计划两个过程。软件发布计划主要是分析用户需求，制定一个大致的计划。而周期开发计划则是前面提到的，XP把复杂的系统划分为多个

验收测试：与瀑布模型不同的时，XP模型的验收测试不是在所有软件功能都完成之后再进行测试，而是用户对每个周期完成时所发布的系统进行评估，这样软件开发对用户来说更具有实际意义，也体现了XP作为一种近螺旋式开发方式的特点。同时，验收测试和测试驱动的开发也保证了各个周期所发布的产品的一致性和可靠性。

小规模发布：即前面提到的，XP模型把复杂的软件开发过程分解为多个周期，将系统分解为多个简单的活动或者任务。因此每个周期开发的需求都是用户需要的东西，XP模型就要求频繁地发布软件，可能的话，应该每天都发布一个新的版本，在对系统某一部分做了修改或者完善之后，也应该立刻发布新版本。

**2. 阅读DevOps文档，了解DevOps**

DevOps（Development和Operations的组合词）是一组过程、方法与系统的统称，旨在促进开发（应用程序/软件工程）、技术运营和质量保障（QA）部门之间的沟通、协作与整合。其核心特点在于通过自动化流程，如软件交付和架构变更，使构建、测试、发布软件更加快捷、频繁和可靠。DevOps的出现源于软件行业对开发与运维之间紧密合作的迫切需求，以确保按时交付高质量的软件产品和服务。

DevOps产品的优势体现在多个方面：

（1）自动化：通过实现持续集成、持续交付和持续部署（CI/CD），DevOps产品可以自动化软件开发、测试和部署流程，减少手动操作，提高工作效率。

（2）敏捷性：DevOps产品支持敏捷开发方法，帮助团队更快地响应需求变化，加快产品迭代周期，提高交付速度。

（3）协作与沟通：DevOps产品促进开发团队、运维团队和其他相关团队之间的合作与沟通，实现跨部门协作，降低信息壁垒，提高工作效率。

（4）可观察性：提供丰富的监控、日志和分析功能，帮助团队了解系统状态、性能指标，及时发现和解决问题。

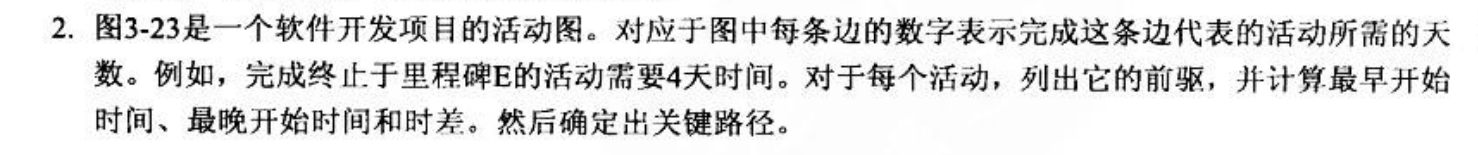
（5）可靠性：通过自动化测试、部署和监控，DevOps产品有助于提高系统稳定性和可靠性，减少故障和停机时间。

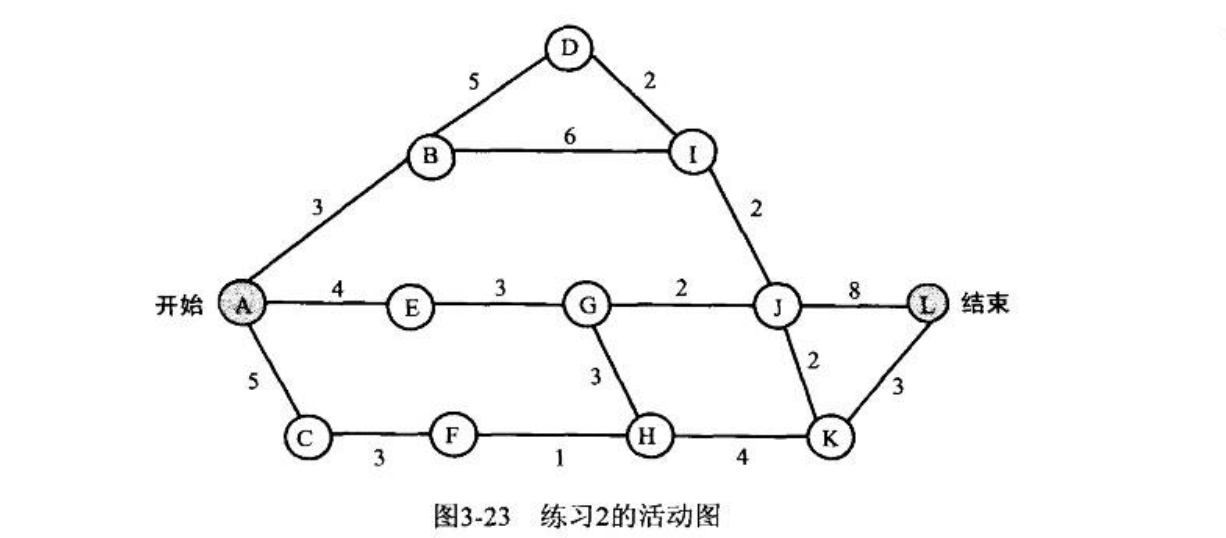
（6）安全性：整合安全审计、漏洞扫描等安全工具，帮助团队及时发现并解决安全问题，加强系统安全性。

在DevOps流程中，通常包括需求分析、编码、构建、测试、部署和反馈等步骤。每个步骤都通过自动化工具和流程实现，确保软件开发的高效性和质量。

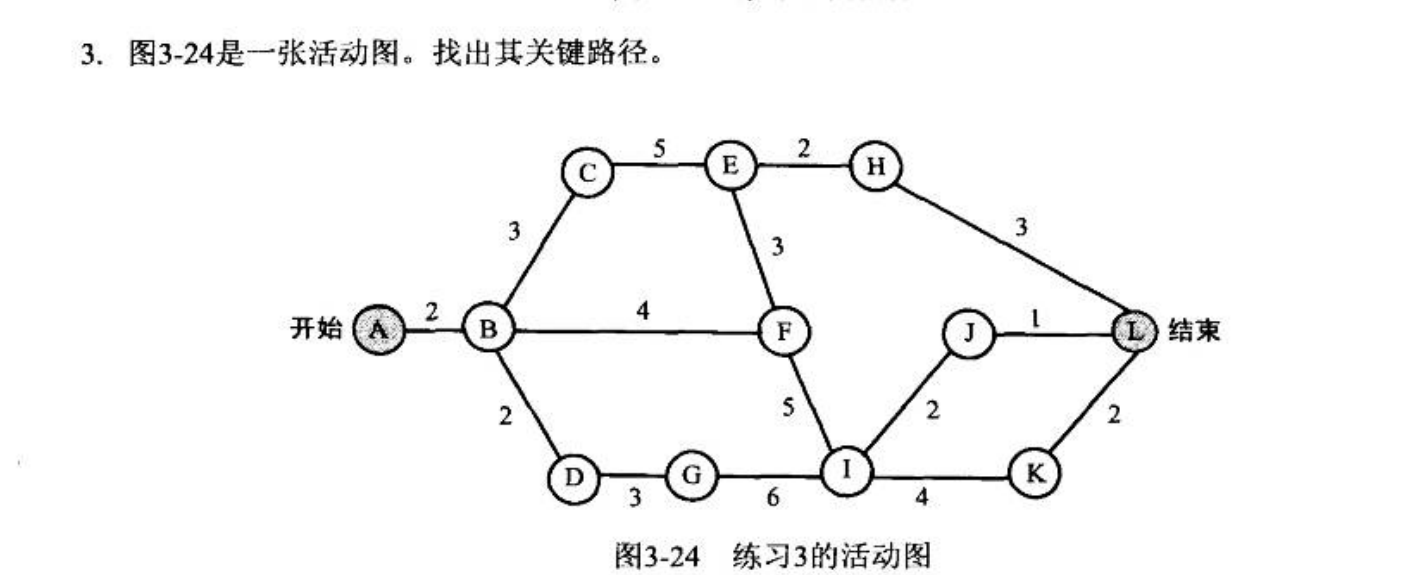
**3. 活动图练习**

**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。**

****

****

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AB** | **AC** | **AE** | **BD** | **BI** | **CF** | **EG** | **FH** | **GH** | **GJ** | **HK** | **IJ** | **JK** | **JL** | **KL** |
| **Ve** | **0** | **0** | **0** | **3** | **3** | **5** | **4** | **8** | **7** | **7** | **10** | **10** | **12** | **12** | **14** |
| **Vl** | **3** | **9** | **5** | **8** | **10** | **12** | **10** | **13** | **13** | **12** | **17** | **12** | **17** | **20** | **20** |
| **权** | **3** | **5** | **4** | **5** | **6** | **3** | **3** | **1** | **3** | **2** | **4** | **2** | **2** | **8** | **3** |
| **E** | **0** | **0** | **0** | **3** | **3** | **5** | **4** | **8** | **7** | **7** | **10** | **10** | **12** | **12** | **14** |
| **L** | **0** | **4** | **1** | **3** | **4** | **9** | **7** | **12** | **10** | **10** | **13** | **10** | **15** | **12** | **17** |
| **时差** | **0** | **4** | **1** | **0** | **1** | **4** | **3** | **4** | **3** | **3** | **3** | **0** | **3** | **0** | **3** |

****

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** |
| **e** | **0** | **2** | **5** | **4** | **8** | **13** | **7** | **12** | **18** | **20** | **22** | **24** |

**关键路径：A-B-C-E-F-I-K-L**

**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。**