

实验四 XP 开发方法，DevOps，活动图

实验目的：

1. 了解 XP 开发方法
2. 了解 DevOps
3. 理解项目活动图

实验内容：

1. 阅读 XP 开发方法文档，理解 XP 过程工作模型

极限编程（Extreme Programming, XP）是一种以提高软件质量和响应能力为目标的敏捷软件开发方法。其主要特点和过程工作模型如下：

结对编程（Pair Programming）：两名程序员共同完成一个任务，一人编写代码，另一人审查，角色定期互换。

测试驱动开发（Test-Driven Development, TDD）：编写代码前先编写单元测试，用测试来驱动开发。

持续集成（Continuous Integration）：频繁将代码集成到主干，尽早发现和解决问题。

重构（Refactoring）：定期重构代码，改善代码结构，提高可维护性。

简单设计（Simple Design）：只实现当前需求的简单设计，避免过度设计。

客户参与（Customer Involvement）：客户实时参与开发过程，提供反馈和需求变更。

小版本发布（Small Releases）：频繁发布小版本，逐步实现完整功能。

40 小时工作周：保持合理的工作强度，避免过度疲劳。

具体内容见 **xp 分析法.md**

2. 阅读 DevOps 文档，了解 DevOps

DevOps 是一种结合软件开发（Development）和 IT 运维（Operations）的方法，通过自动化和持续交付加快软件发布，提高软件质量和可靠性。其主要特点和实践包括：

核心概念

文化：强调团队合作、沟通和共享责任，打破开发和运维之间的隔阂。

自动化：通过自动化工具实现持续集成、持续交付和持续部署，提高效率和质量。

监控与反馈：实时监控系统运行状态，快速响应问题，持续改进。

持续改进：不断优化流程和工具，提高团队的生产力和质量。

实践

持续集成（CI）：频繁将代码集成到主干，并自动化构建和测试。

持续交付（CD）：确保代码在任何时刻都可以安全地部署到生产环境。

基础设施即代码（IaC）：通过代码管理和配置基础设施，实现自动化部署和管理。

监控与日志：监控系统性能和日志，及时发现和解决问题。

协作工具：使用协作工具提高团队沟通和协作效率。

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。

我们小组进行了讨论，， p98 第 2 题，关键路径为 A->B->D->I->J->L，长度为 20

P98 第 3 题,关键路径为 A->B->C->E->F->I->K->L, 长度为 24

我们小组针对自己项目中的工作进行了工作活动分解, 绘制了项目活动图, 并找出了关键路径。

1	学习React和nodejs框架	项目编码
2	数据库模式设计	项目编码
3	用户登录和注册页面	项目编码
4	后端数据库API设计	项目编码
5	导航栏	项目编码
6	边栏	项目编码
7	页面跳转逻辑设计	项目编码
8	商品卡片设计	项目编码
9	个人主页展示	项目编码
10	商品主页	项目编码
11	购买逻辑实现	项目编码
12	搜索和点赞功能	项目编码
13	个人订单处理	项目编码
14	界面美化	项目编码

关键路径: 1 → 2 → 3 → 9 → 13 → 14
总长度: 39

