XP(极限编程); 是一种轻量级的软件开发方法论，属于敏捷开发方法的一种。它的核心思想是通过拥抱变化、快速反馈和团队协作来提高软件开发的效率和质量。

XP的核心特点：

1.核心价值观：

沟通（Communication）：强调团队成员和客户之间的高效沟通。

简单（Simplicity）：只做必要的事情，避免过度设计。

反馈（Feedback）：通过频繁的测试和迭代获取反馈。

勇气（Courage）：敢于面对变化和挑战传统方法。

2. 四个变量：

成本（Cost）：项目所需的资源。

时间（Time）：项目的交付时间。

质量（Quality）：系统的正确性和测试覆盖率。

范围（Scope）：项目需要实现的功能。

3.核心实践：

计划游戏（Planning Game）：通过用户故事卡和迭代计划来管理项目。

小步迭代（Frequent Releases）：通过短周期的迭代快速交付功能。

持续集成（Continuous Integration）：频繁集成代码以减少冲突。

结对编程（Pair Programming）：两名开发者共同编写代码，提高代码质量。

重构（Refactoring）：不断优化代码结构以提高可维护性。

测试驱动开发（Test-Driven Development, TDD）：先写测试，再写代码，确保代码质量。

XP对IT公司人才招聘项目的的好处

1. 关于人才招聘系统的需求可能会随着市场变化而频繁调整，例如新增招聘渠道、优化简历筛选功能等。XP的“拥抱变化”理念允许在项目过程中随时调整需求，通过短周期的迭代快速响应市场变化。
2. 人才招聘系统需要处理大量数据（如简历、候选人信息等），代码质量直接影响系统的稳定性和性能。而XP的测试驱动开发（TDD）：通过单元测试和功能测试确保代码的正确性，和持续集成（CI）：频繁集成代码，及时发现和修复问题，和重构（Refactoring）：优化代码结构，提高可维护性。

3 开发团队成员可能离职或调岗，系统维护需要多人熟悉代码。而XP的结对编程（Pair Programming）：通过结对编程让团队成员熟悉系统各个模块。集体代码所有权（Collective Code Ownership）：任何团队成员都可以修改代码，避免知识孤岛。

DevOps 是一种基于敏捷和精益原则的软件开发方法，强调开发、运维和业务团队之间的协作和集成。

DevOps工作原理：

核心实践：

持续集成（Continuous Integration, CI）：开发人员频繁地将代码合并到共享仓库，并通过自动化测试验证。

持续交付（Continuous Delivery, CD）：确保代码可以随时部署到生产环境。

持续部署（Continuous Deployment）：自动将代码部署到生产环境。

基础设施即代码（Infrastructure as Code, IaC）：通过代码管理基础设施配置。

自动化测试：在每个部署阶段运行自动化测试。

监控和反馈：实时监控系统性能并快速响应问题。

CALMS 模型：

文化（Culture）：促进团队之间的协作和信任。

自动化（Automation）：通过工具减少手动操作。

精益（Lean）：消除浪费，优化流程。

度量（Metrics）：通过数据驱动决策。

共享（Sharing）：开放信息和知识共享。

DevOps对IT公司招聘系统项目的好处？

1.通过自动化构建和部署工具（如 Jenkins、GitLab CI），开发人员可以快速将代码集成到系统中，并自动部署到测试和生产环境。这大大缩短了从开发到部署的时间。

2.开发人员可以频繁地将代码合并到共享仓库，并通过自动化测试验证代码质量。这确保了代码的稳定性和可部署性，减少了手动操作和错误。

3.DevOps 支持快速迭代和持续改进，开发团队可以根据用户反馈快速调整功能和优化系统。

通过工具如 Terraform 或 Ansible，可以将基础设施配置代码化，确保开发、测试和生产环境的一致性，减少因环境差异导致的问题。

4.在每个部署阶段运行自动化测试（单元测试、集成测试、性能测试等），确保系统功能的完整性和性能的稳定性。

5.通过监控工具（如 Prometheus、Grafana）实时监控系统性能和用户行为，及时发现和解决问题，减少系统停机时间

实验五2

先计算VE(最早时间)和VL(最晚时间)的值再推出最早开始时间和最晚开始时间以及时差。

VE[A]=0,VE[B]=1,VE[C]=7,VE[D]=10,VE[E]=2,VE[F]=4,VE[H]=[15,8,11]->15,VE[G]=MAX[6,9,14,15]->15,VE[I]=18,VE[J]=24,VE[K]=26,VE[L]=24,VE[M]=18,VE[N]=MAX[28,25]->28,VE[O]=[34,36,38]->38,VE[Q]=32,VE[P]=[47,34]->47。

计算VL的值

VL[P]=47,VL[O]=38,VL[Q]=45,VL[N]=min[28,41]->28,VL[M]=21,VL[L]=24,VL[H]=min[19,15]->15,VL[K]=36,VL[J]=29,VL[I]=28,VL[G]=[25,22]->22,VL[B]=17,VL[C]=20,VL[D]=min[18,10]->10,VL[E]=9,VL[F]=8,VL[A]=min[16,13,0,7,4]=0.

|  |  |  |
| --- | --- | --- |
| 节点 | VE(最早时间 | VL(最晚时间) |
| A | 0 | 0 |
| B | 1 | 17 |
| C | 7 | 20 |
| D | 10 | 10 |
| E | 2 | 9 |
| F | 4 | 8 |
| G | 15 | 22 |
| H | 15 | 15 |
| I | 18 | 28 |
| J | 24 | 29 |
| K | 26 | 36 |
| L | 24 | 24 |
| M | 18 | 21 |
| N | 28 | 28 |
| O | 38 | 38 |
| Q | 32 | 32 |
| P | 47 | 47 |

计算ES(最早开始时间)和LS（最晚开始时间的值），以及时差Slack

|  |  |  |  |
| --- | --- | --- | --- |
| 每个活动 | ES | LS | Slack |
| A->B | 0 | 16 | 16 |
| A->C | 0 | 13 | 13 |
| A->D | 0 | 0 | 0 |
| A->E | 0 | 7 | 7 |
| A->F | 0 | 4 | 4 |
| B->G | 1 | 7 | 6 |
| C->G | 7 | 20 | 13 |
| D->G | 10 | 18 | 8 |
| D->H | 10 | 10 | 0 |
| E->H | 2 | 9 | 7 |
| F->H | 4 | 8 | 4 |
| H->G | 15 | 21 | 6 |
| G->I | 15 | 25 | 10 |
| G->J | 15 | 20 | 13 |
| I->O | 15 | 28 | 6 |
| J->K | 24 | 31 | 7 |
| K->O | 26 | 36 | 10 |
| H->L | 15 | 15 | 0 |
| H->M | 15 | 18 | 3 |
| L->N | 24 | 24 | 0 |
| M->N | 18 | 21 | 3 |
| N->O | 28 | 28 | 0 |
| O->P | 38 | 38 | 0 |
| N->Q | 28 | 28 | 0 |
| Q->P | 32 | 45 | 13 |

关键路径：找出时差为0的活动。

A->D,D->H,H->L,L->N,N->O,O->P,N->Q

其中A->D-H->L->N->O->P构成关键路径，总长度为：10+5+9+4+10+9=47.