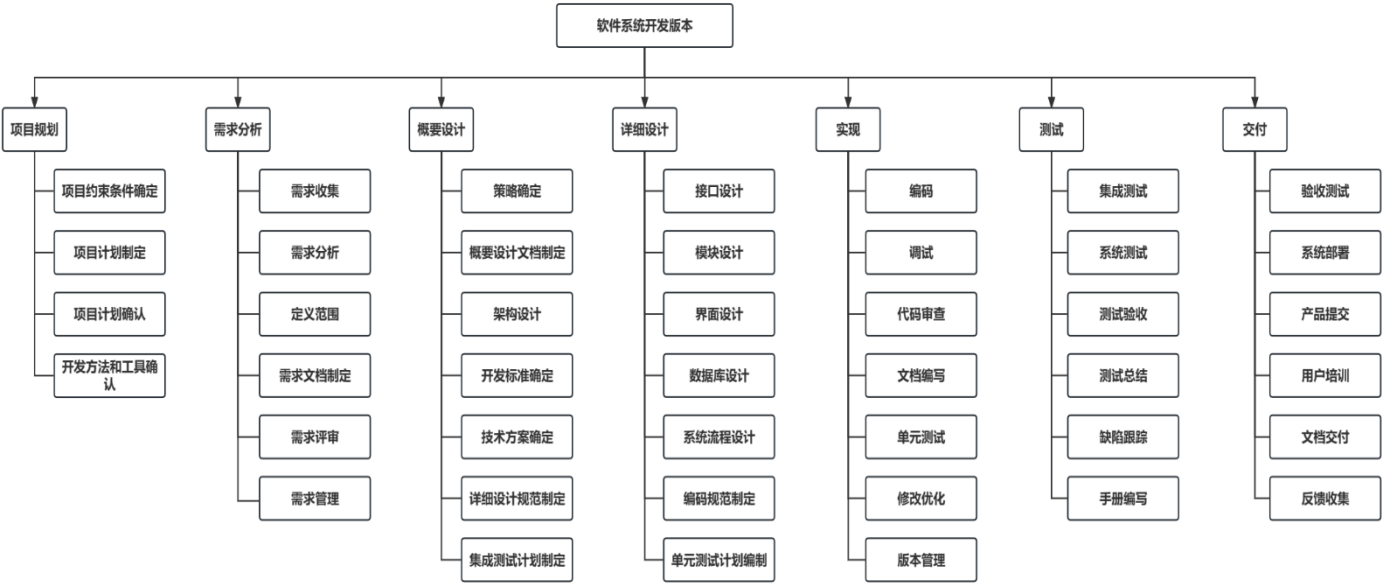


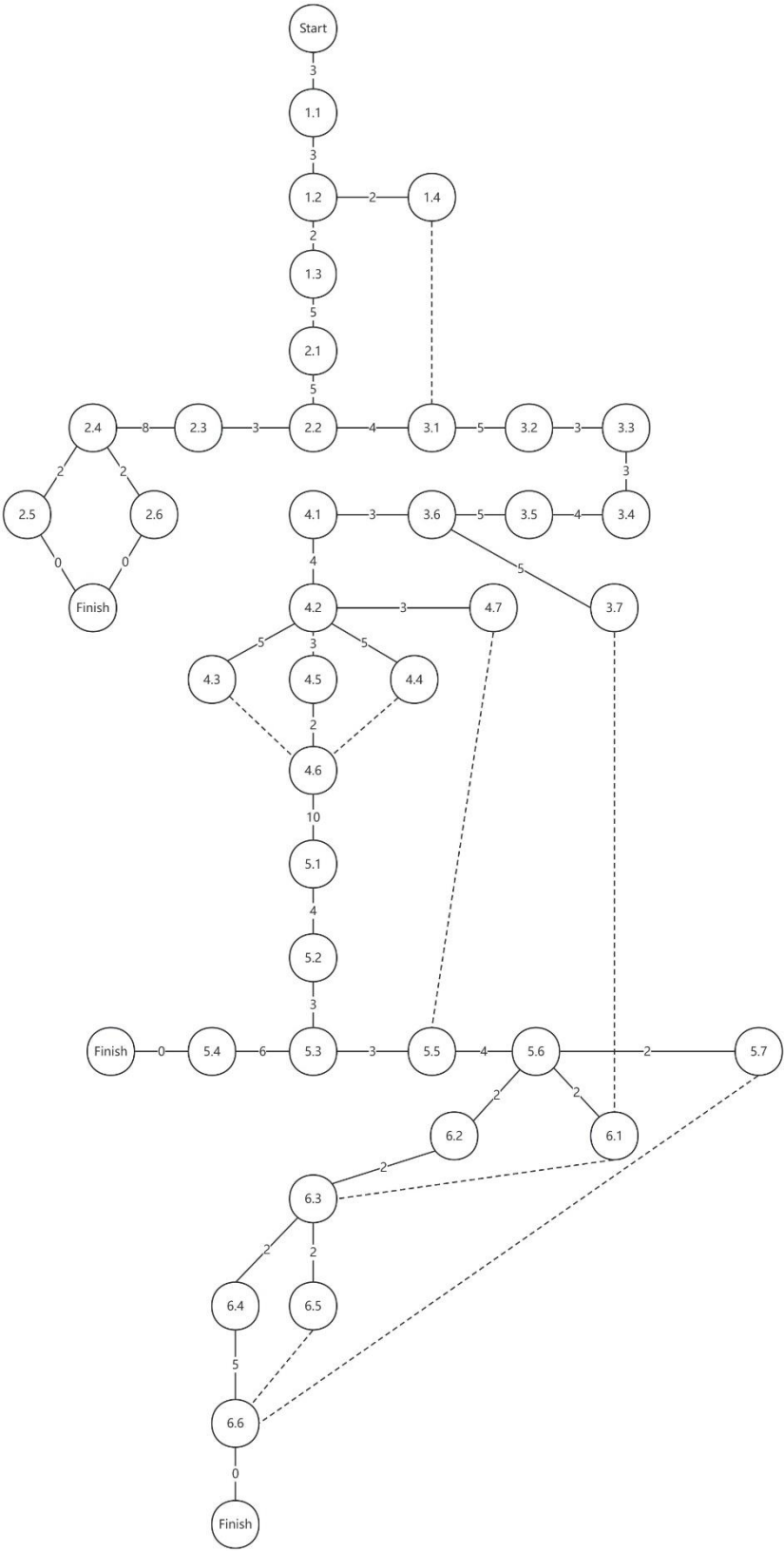
活动图

1 活动图分解、绘制与关键路径

1.1 WBS



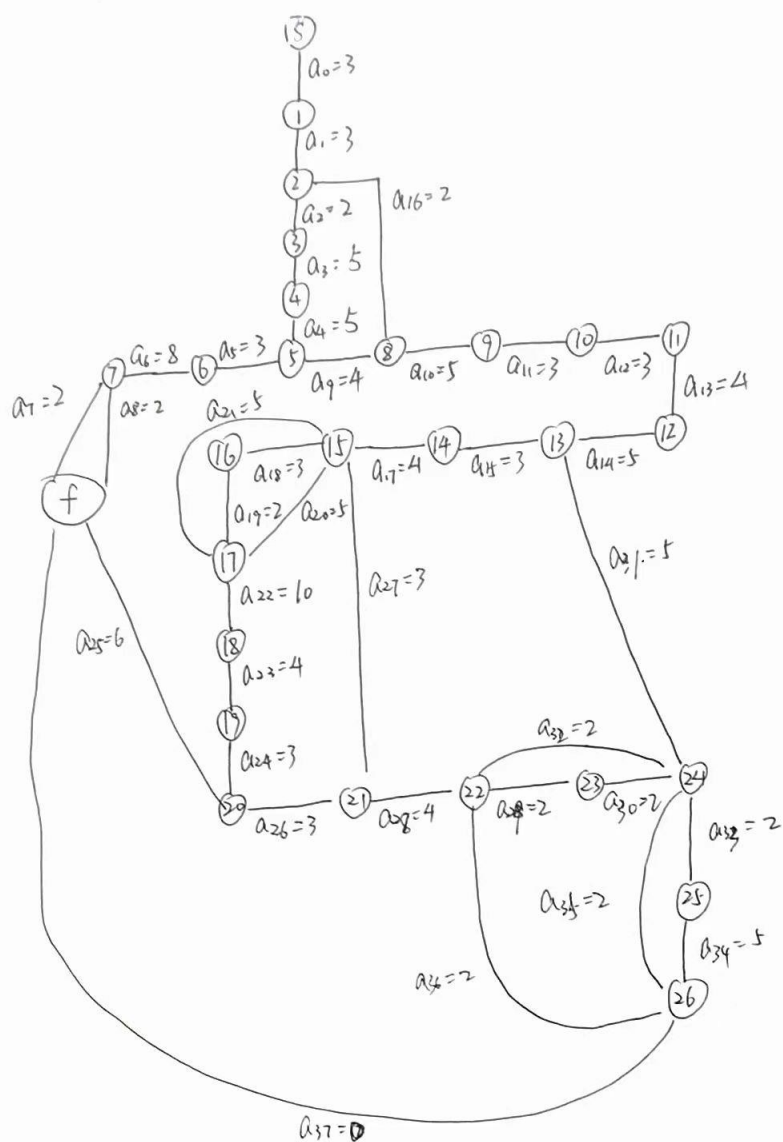
1.2 工作分解结构与活动图



1.3 进展估计图

任务	预计完成时间（天）
步骤 1：项目规划	
任务 1.1：项目约束条件确定	3
任务 1.2：项目计划制定	3
任务 1.3：项目计划确认	2
任务 1.4：开发方法和工具确认	2
步骤 2：需求分析	
任务 2.1：需求收集	5
任务 2.2：需求分析	5
任务 2.3：定义范围	3
任务 2.4：需求文档制定	8
任务 2.5：需求评审	2
任务 2.6：需求管理	2
步骤 3：概要设计	
任务 3.1：策略确定	4
任务 3.2：概要设计文档制定	5
任务 3.3：架构设计	3
任务 3.4：开发标准确定	3
任务 3.5：技术方案确定	4
任务 3.6：详细设计规范制定	5
任务 3.7：集成测试计划制定	5
步骤 4：详细设计	
任务 4.1：接口设计	3
任务 4.2：模块设计	4
任务 4.3：界面设计	5
任务 4.4：数据库设计	5
任务 4.5：系统流程设计	3
任务 4.6：编码规范设计	2
任务 4.7：单元测试计划编制	3
步骤 5：实现	
任务 5.1：编码	10
任务 5.2：调试	4
任务 5.3：代码审查	3
任务 5.4：文档编写	6
任务 5.5：单元测试	3
任务 5.6：修改优化	4
任务 5.7：版本管理	2
步骤 6：测试	
任务 6.1：集成测试	2
任务 6.2：系统测试	2
任务 6.3：测试验收	2
任务 6.4：测试总结	2
任务 6.5：缺陷跟踪	2
任务 6.6：手册编写	5

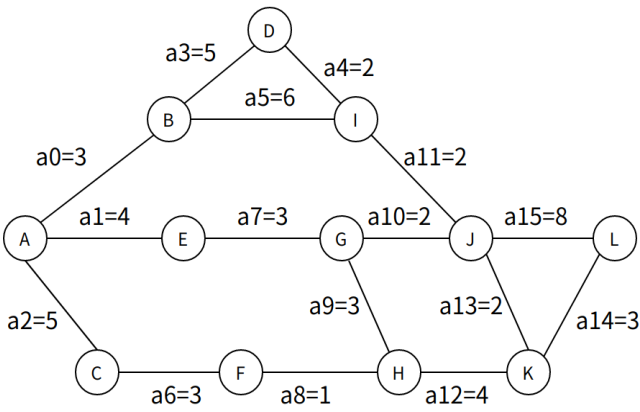
1.4 关键路径



关键路径: $\langle a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{17}, a_{18}, a_{19}, a_{22}, a_{23}, a_{24}, a_{26}, a_{28}, a_{29}, a_{30}, a_{33}, a_{34}, a_{37} \rangle$

2 活动图-课本练习题

2.1 题 2



前驱节点：

	A	B	C	D	E	F	G	H	I	J	K	L
前驱	NULL	A	A	B	A	C	E	G	D	I	H	J

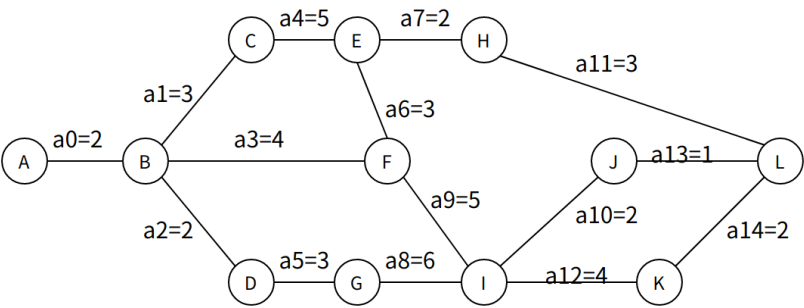
最早和最晚开始时间以及时差：

项目	A	B	C	D	E	F	G	H	I	J	K	L
earliest(i)	0	3	5	8	4	8	7	10	10	12	14	20
latest(i)	0	3	9	8	7	12	10	13	10	12	17	20

项目	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15
early(k)	0	0	0	3	8	3	5	4	8	7	10	10	10	12	14	12
late(k)	0	3	4	3	8	4	9	7	12	10	10	10	13	15	17	12
时差	0	3	4	0	0	1	4	3	4	3	3	0	3	3	3	0
	*			*	*							*				*

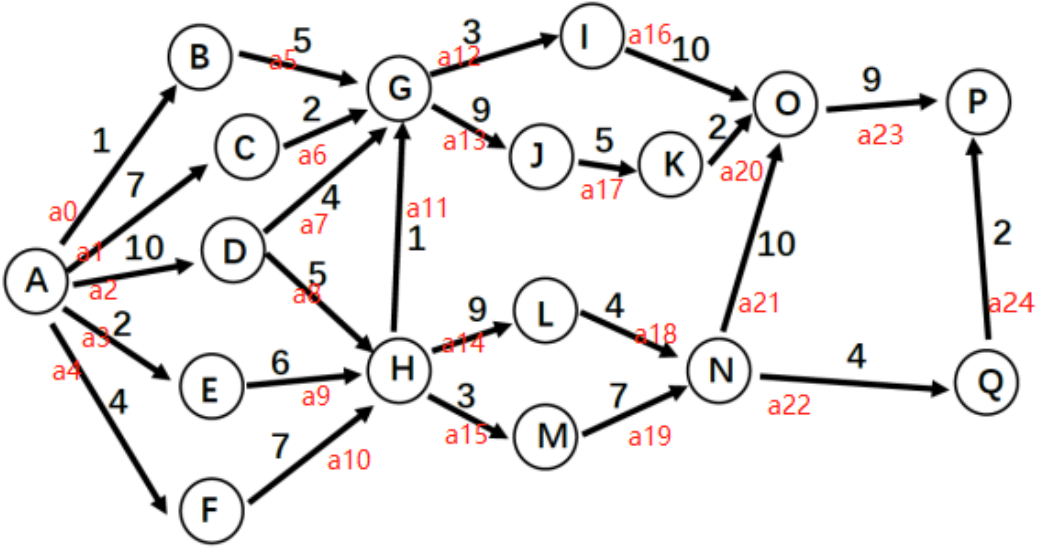
关键路径为 a0→a3→a4→a11→a15

2.2 题 3



关键路径为 a0→a1→a4→a6→a9→a12→a14

3 活动图-课件练习题

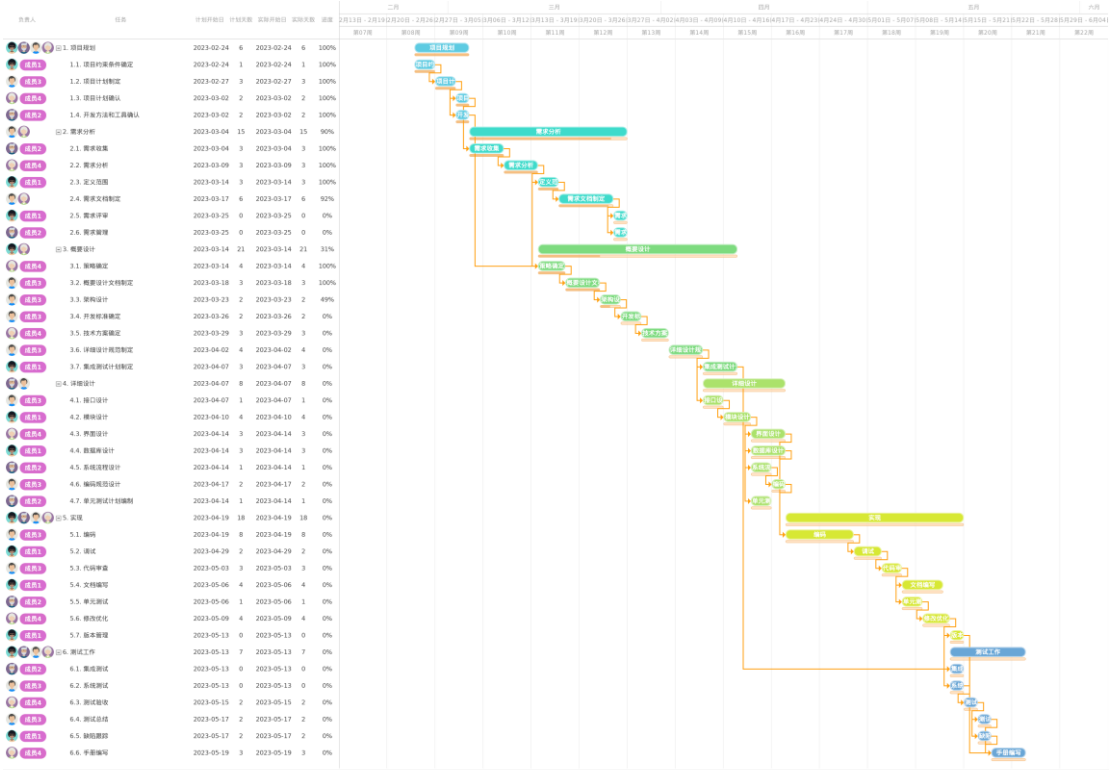


项目	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q										
earliest(i)	0	1	7	10	2	4	16	15	19	25	30	24	18	28	38	47	32										
latest(i)	0	17	19	10	9	8	22	15	28	31	36	24	21	28	38	47	45										
项目	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24		
early(k)	0	0	0	0	0	1	7	10	10	2	4	15	16	16	15	15	19	25	24	18	30	28	28	38	32		
late(k)	14	11	0	7	4	17	20	18	10	9	8	21	25	22	15	18	28	31	24	21	36	28	41	38	45		
时差	14	11	0	7	4	16	13	8	0	7	4	6	9	6	0	3	9	6	0	3	6	0	13	0	13		
关键路径			*						*						*			*			*		*		*		

关键路径为 a2→a8 →a14→a18→a21→a23

总长度：6

4 甘特图记录跟踪项目



5 调研

5.1 国内外软件开发调研

5.1.1 调研国内外软件开发团队组织结构和工作方式对比

1. 组织结构

在国内，软件开发团队一般由项目经理、技术负责人、开发人员和测试人员组成。项目经理负责整个项目的进度和管理，技术负责人负责技术选型和架构设计，开发人员负责编码和测试人员负责测试和验收。开发团队的领导结构比较平坦，每个人员的职责比较清晰。

在国外，软件开发团队一般由产品经理、技术负责人、开发人员、测试人员、运维人员等组成。产品经理负责需求分析和产品规划，技术负责人负责技术选型和架构设计，开发人员负责编码，测试人员负责测试和验收，运维人员负责部署和维护。开发团队的领导结构比较扁平化，每个人员的职责较为灵活，可以相互补充。

2. 工作方式

在国内，软件开发团队的工作方式比较注重执行力和效率。开发人员和测试人员一般都需要按照项目计划完成各自的任务，进度和质量比较重要。在开发过程中，一般会有比较严格的代码审查和测试流程，确保代码质量和产品质量。

在国外，软件开发团队的工作方式比较注重创新和用户体验。产品经理和技术负责人会根据市场需求和用户反馈不断调整产品规划和技术选型，开发人员和测试人员需要不断探索新的技术和方法，提升用户体验。在开发过程中，一般会有比较宽松的代码审查和测试流程，鼓励开发人员自主创新和探索。

3. 沟通协作

在国内，软件开发团队的沟通协作比较注重文档和会议。开发人员和测试人员一般需要编写详细的设计文档和测试报告，并定期参加各种会议，如需求确认会、设计评审会、测试验收会等，确保各方都了解项目进展和质量。

在国外，软件开发团队的沟通协作比较注重在线协作和团队文化。开发人员和测试人员一般会使用各种在线工具，如 **Slack**、**Zoom** 等，进行即时沟通和协作。团队文化也非常重要，鼓励团队成员之间相互帮助、分享经验和知识，以及不断探索新的技术和方法。

5.1.2 调研国内与国外软件开发团队的管理方式对比

1. 团队组织结构

在国内，软件开发团队的组织结构相对比较简单，通常分为产品经理、技术负责人、开发人员和测试人员等几个角色。产品经理负责确定产品的需求和规划，技术负责人则负责技术架构和技术选型，开发人员编写代码，测试人员则进行测试和质量控制。这种组织结构的优点是明确分工、效率高，但缺点是沟通不够紧密，协作不够灵活。

在国外，软件开发团队的组织结构相对复杂一些。除了上述角色外，还会有 UI/UX 设计师、DevOps 工程师、运维工程师、安全工程师等不同的角色。同时，还会有技术咨询师、项目经理等人员参与到团队中，共同制定和实施开发计划。这种组织结构的优点是角色分工明确、沟通协作更加紧密，缺点是需要耗费更多的人力和时间来管理团队。

2. 沟通与协作

在国内，由于团队规模较小，沟通和协作相对简单。一般采用传统的面对面交流、邮件和即时通讯等方式来沟通和协作。缺点是沟通效率较低，信息传递不够及时和准确。

在国外，由于团队规模较大，沟通和协作更加复杂。为了提高沟通效率和协作质量，通常采用更加高效的协作工具和流程。例如，使用 Slack、Microsoft Teams、Zoom 等即时通讯和视频会议工具来进行沟通；使用 Jira、Trello 等项目管理工具来协调工作流程；使用 Git、GitHub 等版本控制工具来协作开发。同时，还会通过代码审查、持续集成等方式来保证代码质量和项目进度。这种沟通和协作方式的优点是效率高、信息传递准确，缺点是需要投入更多的资源和时间来实现。

3. 文化

国内和国外软件开发团队的文化也存在一定的差异。在国内，软件开发团队的文化主要是执行力和效率导向。团队成员之间的关系通常比较简单，注重个人责任和任务完成。在企业中，晋升和加薪往往基于员工的工作表现，而不是基于员工的潜力和能力。

在国外，软件开发团队的文化更加注重创新、团队精神和平等。团队成员之间的沟通和协作更加开放和平等，注重团队文化和价值观的传承和发扬。同时，员工的晋升和加薪也更加注重员工的技能和潜力，而不仅仅是他们的工作表现。

4. 工作流程

在国内，软件开发团队的工作流程通常比较简单。产品经理提出需求，技术负责人确定技术方案，开发人员进行开发，测试人员进行测试，最后发布产品。整个过程通常采用瀑布式开发模型，按照固定的流程进行。

在国外，软件开发团队通常采用敏捷开发模型，工作流程更加灵活。产品经理、技术负责人、开发人员和测试人员等角色之间的沟通更加紧密，通过持续集成和持续交付等方式来快速迭代和发布产品。同时，团队成员之间的反馈和交流也更加频繁和及时，以保证产品的质量和用户体验。

5.2 工作调研

5.2.1 从个人角度，你最喜欢的工作方式、工作环境条件、可接受的约束等是什么

作为团队成员之一，我最喜欢的工作方式是敏捷开发。敏捷开发强调团队合作、快速反馈和持续改进，可以帮助团队成员更好地了解项目需求和目标，并且在项目过程中快速地响应变化和不断改进产品。

在工作环境条件方面，我希望能够在安静且宽敞的工作环境中工作。在这样的环境中，我可以更专注地完成工作，并且可以避免干扰和噪音的影响，从而提高工作效率。

在可接受的约束方面，我认为在项目进度方面的约束是可以接受的。我认为每个人都应该尽力按时完成自己的任务，避免延迟项目进度，并确保团队的共同目标得以实现。但是，我也认为应该为团队成员提供足够的时间来完成任务，避免过度加班和压力。

5.2.2 从团队项目管理角度，你认为最有效的项目组工作管理方式是什么

敏捷开发。

1. 敏捷开发适用于小型项目：对于小型博客平台项目，需求变化频率相对较低，不需要过多的计划和文档管理。敏捷开发的迭代式开发方法可以使开发团队更快速地响应变化和快速交付产品。
2. 敏捷开发注重团队合作：敏捷开发中的团队合作和沟通非常重要，这可以帮助团队成员更好地了解项目需求和目标，同时可以减少沟通成本和防止信息偏差。
3. 敏捷开发强调快速反馈和持续改进：敏捷开发强调快速反馈和持续改进，可以让开发团队在整个项目周期中不断地检查和改进产品。这可以帮助团队快速识别问题并解决问题，确保产品质量和用户满意度。
4. 敏捷开发适用于小团队：敏捷开发可以适用于小团队，每个人都有明确的角色和职责，可以有效地协作完成项目。