



WORK LOG 2

Shandong University

March 8, 2020

高德琛

Contents

1	引言	3
1.1	日志编写目的	3
1.2	日志编写逻辑	3
2	敏捷开发调研	4
2.1	概要	4
2.2	敏捷开发宣言	4
2.3	敏捷开发背后的原则	4
2.4	敏捷开发的概念	5
2.4.1	价值观、原则	5
2.4.2	以用户需求为核心	5
2.4.3	大项目划分为小项目	5
2.5	敏捷开发的特征	5
2.5.1	迭代式开发	5
2.5.2	增量交付	6
2.5.3	开发团队和用户反馈推动产品开发	6
2.5.4	持续集成	6
2.5.5	开发团队自我管理	6
2.6	敏捷开发原则	6
2.6.1	快速迭代	6
2.6.2	需求评审	6
2.6.3	编写 story、验收标准	7
2.6.4	多沟通	7
2.6.5	做好原型	7
2.6.6	及早考虑测试	7
2.7	方法论-Scrum	7
2.7.1	Scrum 开发流程的三大角色	7
2.7.2	团队的 Scrum 实施过程	8

3	敏捷开发的适应性	9
3.1	以“人”为核心	9
3.2	对需求变更的认知	9
3.3	研发工作的量化	10
3.4	为提升而重构	10
3.5	减负的艺术	10
4	项目设计工作	11
4.1	工作总结	11
4.2	功能模块划分	11
4.3	用例图	12
4.4	开销的预估	12
4.4.1	一期开销	13
4.4.2	二期开销	13

1 引言

1.1 日志编写目的

《Work Log 2》收集了个人在本此实验当中的工作记录。比起记录，该文档更大的意义在于它可以是计划性的；只有当文档中预定的工作任务完成，才能称之为“记录”。

软件工程的原则告诉我们“文档先行”的重要性。软件开发本质是人与人的合作，其主体是活生生的“人”。而不同的人自然会有不同的开发理念、习惯。为了在规约上达成一致，形成量化的默契，“文档先行”可以最小化团队的沟通成本。同样地，个人的计划和工作日志也是富有价值的。

通过本次工作日志，得以整理罗列所学原理之间的逻辑，使之条理清晰、重点分明；从而让软件开发原理在小组讨论当中发挥更大的价值。

1.2 日志编写逻辑

由于我的主要工作是调研敏捷开发，进一步推进项目设计。所以本日志首先在章节 2 介绍敏捷开发相关原理的调研，侧重方法论；继而在章节 3 介绍个人对于敏捷开发的理解、相关价值理念、以及对于项目的普适性探究；最后章节 4 包含个人在项目设计当中的调研和日志。

2 敏捷开发调研

2.1 概要

本章节旨在整理敏捷开发原理，用于小组讨论原料。个人对敏捷开发的开发将在章节 3 中详述。

2.2 敏捷开发宣言

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software **over** comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, **while there is value in the items on the right**, we value the items on the left more.

2.3 敏捷开发背后的原则

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals.

Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity—the art of maximizing the amount of work not done—is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2.4 敏捷开发的概念

2.4.1 价值观、原则

敏捷开发是一种价值观与原则，指导我们更加高效的开发。

2.4.2 以用户需求为核心

敏捷开发以用户需求为核心，采用迭代 (时间周期)、增量 (循序渐进，功能模块) 的方式开发软件，目的在于快速覆盖、响应市场需求。

2.4.3 大项目划分为小项目

大项目划分为小项目，分别完成，独立运行，如微服务的开发过程，就是将系统独立进行开发。

2.5 敏捷开发的特征

2.5.1 迭代式开发

项目按照时间周期进行迭代，比如 A 功能优先级比较高，则在第一个迭代周期内优先开发 A 功能，并上线。第二个迭代周期开发 B 功能。

2.5.2 增量交付

代表产品是在每个周期结束时被逐步交付使用的。如微软在吸取 Vista 操作系统采用传统的瀑布式开发流程之后，发现操作系统并不能完全覆盖用户的需求。在 2005-2007 两年时间内通过内部推行的敏捷开发原则，上线了 Windows 7，获得市场的一致好评。

2.5.3 开发团队和用户反馈推动产品开发

敏捷开发提倡用户参与到产品或项目开发的整个流程当中，通过用户反馈使得产品更加符合用户频繁变动的需求。

2.5.4 持续集成

采用敏捷开发的产品在产品初期会上线基本功能，之后的功能是根据收集到的用户反馈进行开发的，实现功能模块的持续集成。

2.5.5 开发团队自我管理

传统的开发模式，注重文档约束，而敏捷开发原则的推行原则要求团队内部交流便利、文化相对开发，除去必要的文档约束，如 Api 接口文档，最注重的是团队成员的高效交流，以此来提高产品、项目的开发效率、开发质量。

2.6 敏捷开发原则

2.6.1 快速迭代

小版本更新发布，更快覆盖当前市场、用户需求。

2.6.2 需求评审

需求评审阶段，要求 PM、所有相关开发人员参与到需求评审当中。

需求评审阶段：需求可行性分析、确定需求功能范围、PM 对需求中存在异议的细节进行解释。

2.6.3 编写 **story**、验收标准

PM 编写 **story**、验收标准

2.6.4 多沟通

PM、开发人员之间需要多沟通、减少不必要的文档。

2.6.5 做好原型

需求评审完毕后，PM 与 UE、UI 人员进行紧密沟通，完成指导开发人员开发的 UE、UI。

2.6.6 及早考虑测试

测试人员在这个阶段需要根据需求中划分的功能点，设计测试用例。

2.7 方法论-**Scrum**

敏捷开发的方法有很多：ASD、AUP、DSDM、XP、FDD、Kanban、RAD、Scrum。目前国内最流行的应属 Scrum。

Scrum 的英文意思是橄榄球运动的一个专业术语，表示“争球”的动作；把一个开发流程的名字取名为 Scrum，寓意开发团队在开发一个项目时，大家像打橄榄球一样迅速、富有战斗激情、人人你争我抢地完成它。而 Scrum 就是这样的一个开发流程，运用该流程旨在使得团队高效运作。

2.7.1 **Scrum** 开发流程的三大角色

1. 产品负责人 (Product Owner): 主要负责确定产品的功能和达到要求的标准，指定软件的发布日期和交付的内容，同时有权力接受或拒绝开发团队的工作成果。
2. 流程管理员 (Scrum Master): 主要负责整个 Scrum 流程在项目中的顺利实施和进行，以及清除挡在客户和开发工作之间的沟通障碍，使得客户可以直接驱动开发。
3. 开发团队 (Scrum Team): 主要负责软件产品在 Scrum 规定流程下进行开发工作，人数控制在 5 10 人左右，每个成员可能负责不同的技术方面，但要求每成

员必须要有很强的自我管理能力，同时具有一定的表达能力；成员可以采用任何工作方式，只要能达到 **Sprint** 的目标。

2.7.2 团队的 **Scrum** 实施过程

考虑到团队采用敏捷开发的理念不可能一蹴而就，需要循序渐进的让我们组员接受 **Scrum** 方法，为此我们罗列出可行的制度实施流程。

1. 引入任务可视化：用一块白板，让团队成员把所有的开发任务都用便签写好放上去，也可以找一个专业的看板工具。团队中每个人的工作都会被所有人看到，会形成一种无形的监督。
2. 实施站会制度：当团队所有人的任务都在看板后，这些任务的状态是需要有人更新的。无论是专人更新，还是每个人自己更新，都会出现忘记更新，更新不及时的情况。这时，就可以顺势引入站会制度了。每次小组讨论的时候，所有人都花上 15 分钟，说一下自己做了什么，准备做什么，所有人一起把看板更新了。
3. 引入回顾会议：当站会制度成熟后，就可以挑一个版本发布的日子。等版本发布后，把团队成员叫到一起，开一个回顾会。让团队成员对这个版本开发的情况做个总结，同时可以让大家用便签把这个开发版本遇到的问题、不满写下来。针对这些问题不满，大家一起讨论一些改进方案。
4. 引入计划会议：如果再開回顾会的时候，有人提出需求不清，拆解不够细致，工时估算不准之类的问题，那你真是太幸运了。碰到这种情况，你顺势提出这个制度：每个版本开发前，产品经理要和团队人员开个会，一起逐条对需求。产品经理要向团队解释每个需求，并且对团队提出的疑问进行解释和澄清。开发团队要将需求分解成任务，估算工时，最终形成一个开发清单。
5. 引入评审会议：当计划会成功开过几次之后，就可以考虑引入评审会了。你可以在某个重大功能发布的时候，把所有项目相关的人员召集起来，向大家展示团队做了哪些了不起的功能，也可以和大家讨论讨论还可以做哪些优化。
6. 引入 **Scrum**：当整个流程都跑顺以后，就可以正式引入 **Scrum**。此时需要任命 PO、SM，对团队培训 **Scrum** 知识，讲解 **Scrum** 的价值观。

3 敏捷开发的适应性

3.1 以“人”为核心

章节 2.2 当中值得强调的是，作者并没有否认 **right value**，而是侧重于 **left value**。也就是说，尽管比其文档而言，敏捷开发看重实际代码，但是：文档一样重要。敏捷开发的转移对传统工程中繁重部分的注意力，而致力于确实推进软件项目的部分。

敏捷宣言原则与传统软件开发方法很明显的不同在于，就是让我深刻感受到了工程项目中的“人”的元素。软件开发是“人与人的合作”，软件开发的目的是满足“人的需求”（章节 2.3 中倡导“欣然接受需求变更”）。一切都是为了活生生的人服务，所以我们在敏捷开发当中促进开发者的有效沟通交流、尽可能地满足人的需求。

3.2 对需求变更的认知

发掘“人”的元素，尽管合乎情，但是我个人理解当中，这类主观性元素似乎难以合乎“理”。正因为不同的人具有不同的见解，潜意识当中具有不同的标准和规约，所以软件开发当中对于用户需求的标准也难以具体量化，故而往往难以衡量我们的项目，是否真正的“合乎情理”，确切的满足用户的需求。

最为直接的方法，就是通过一次次上线测试和反馈收集，去增量以及迭代我们的产品。而每次增量和迭代的代价似乎过于高，为了满足当下市场快速变化的需求，我们必须尽快将新产品推向市场，快速的实验。

就满足“需求”而言，新的市场环境要求开发者能够更深入的思考，如何去满足用户需求。这里列出三个具有代表性的问题。

1. 如何做出真正满足用户需求的产品？
2. 如何满足不断变化的需求？
3. 如何满足不同层次用户的需求？

经过讨论我们发现，为此而生的敏捷开发理念当中对上述问题有着深入的见解和指导方向。具体而言，敏捷开发体现了早期交付、风险降低的优势。

比如说，按照瀑布模型，每一阶段都依赖前一阶段，造成了交付延迟高。这对于如今市场需求日新月异的格局下，不适用于许多周期长、需求多样的项目。而敏捷开发能够早期交付原型产品，这就为项目带来了初步市场反馈、初步的现金流，有助于团队即时了解市场变化，及时调整迭代方向，帮助团队发展形成良性循环。

3.3 研发工作的量化

研发工作往往是比较难以量化的，对于研发管理来说，即使把迭代周期压缩到了 1-2 周，对于进度的控制粒度也是远远不够的。我一般会要求团队从第一天开始就尽可能的实现持续集成（每日构建），哪怕产品啥都没有，不仅仅是工程团队，产品所有相关的人从第一天开始就可以真实的运行产品，这比什么都更直观的反应了产品进度到哪里。

在持续集成中还有很多相关的规范需要团队执行，比如自动化测试，对于一些核心代码尤为重要，有了自动化测试的用例覆盖，对于每天构建的版本质量就有了更清晰和量化的数据。自动构建的结果会每天发送邮件的团队所有人，每个人都可以清楚了解到项目的进展情况。

3.4 为提升而重构

在我们的开发过程中，重构是无处不在的，这甚至是一种生活方式。对于代码的坏味道，我们要零容忍，并且鼓励团队的成员去阅读优秀的代码，通过重构让代码走在变好的路上。

可能有人会觉得重构降低了开发速度，但其实不是，当代码结构越来越复杂，产品功能越来越多，重构的必要性就越来越大。我在现实的工作中，就发现有不少的案例，比如一个函数改到后来，可能函数名和实际代码完成的工作已经完全不一致了，这种代码，不但非常影响其他人的阅读理解，还经常造成 Bug，甚至会到导致后续功能扩展时花费大量的时间。又比如，现在的前端开发技术，已经发展得非常不同，可能若干年前，我们还在使用 jQuery，还没有组件化的概念，而我现在的创业团队，已经在大规模使用 React and TypeScript 进行构建了。新技术带来的好处是非常明显的，既安全，可扩展性也非常棒。

3.5 减负的艺术

合理的使用工具，可以让团队沟通更顺畅，管理成本也随之降低，同时，我们尽可能的让工作流程简单，去除一切没有必要的环节。

4 项目设计工作

4.1 工作总结

本章节介绍本周个人对项目设计工作的工作记录。包括对功能模块的初步划分、用例图的初步设计、预算的初步估计。

4.2 功能模块划分

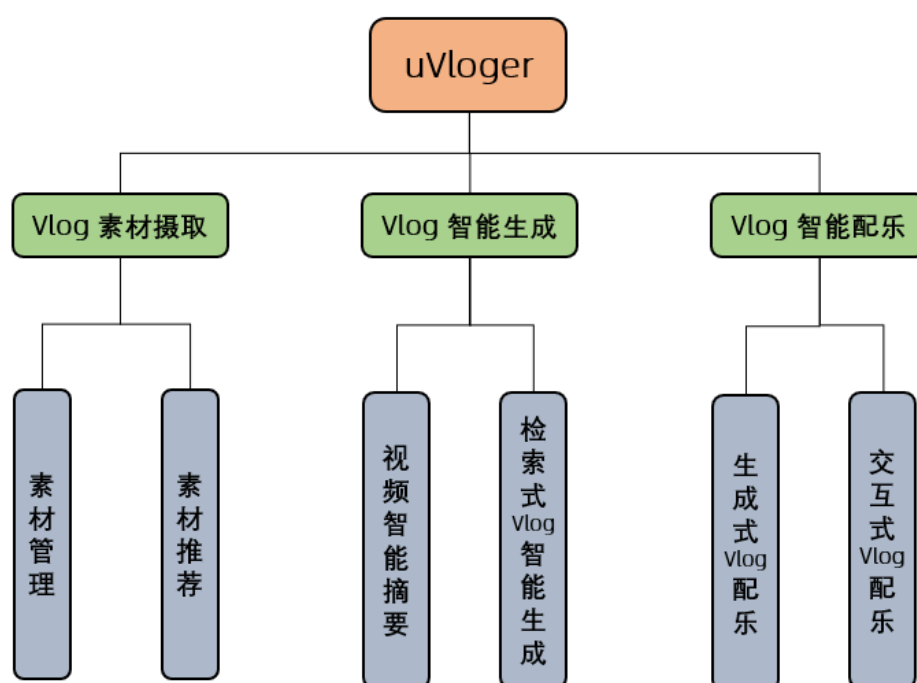


图 1: 功能模块图

4.3 用例图

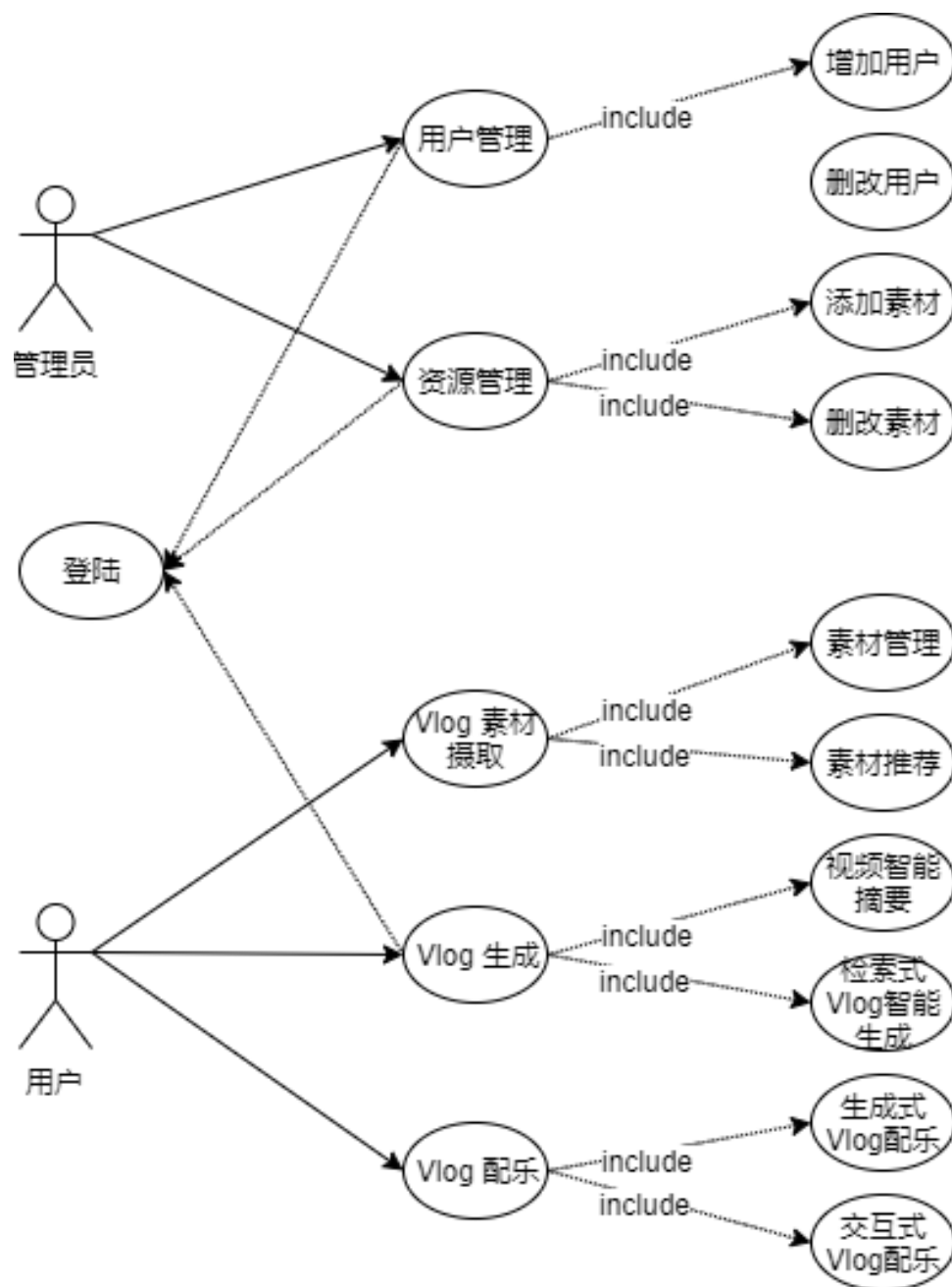


图 2: 用例图

4.4 开销的预估

经过讨论我们按时序和必要性将开销划分为一期、二期。一期开销代表最近一次迭代中必然的花费，二期代表计划中的需求扩张可能带来的花费。

4.4.1 一期开销

开销项目	用途	开销金额
OBS 对象存储服务-存储空间	存储数据集	45.64 元
OBS 对象存储服务-流量	数据集传输	254.10 元
移动硬盘	本地存储数据集	420.00 元
ECS 云服务器	数据集下载及预处理	345.00 元
合计	1064.74 元	

4.4.2 二期开销

开销项目	用途	开销金额
内存型 ECS 服务器	软件测试，部署内存型数据库	150.00 元
CPU 型 ECS 服务器	服务器调试	250.00 元
	服务器软件测试	400 元
	深度学习模型训练，数据集预处理	500 元
GPU 型 ECS 服务器	深度学习模型训练	2700 元
合计	4000 元	