



TEAM WORK LOG 2

Shandong University

March 8, 2020

CODE & NOTE

Contents

1	引言	3
1.1	日志编写目的	3
1.2	日志编写逻辑	3
2	V 模型	4
2.1	V 模型定义	4
2.2	V 模型流程	4
2.2.1	需求分析	4
2.2.2	概要设计	5
2.2.3	详细设计	5
2.2.4	软件编码	5
2.2.5	单元测试	5
2.2.6	集成测试	5
2.2.7	系统测试	5
2.3	验收测试	6
3	顺序生命周期模型	6
4	渐进开发生命周期模型	6
5	迭代生命周期模型	7
5.1	模型定义	7
5.2	迭代周期	8
5.2.1	需求分析阶段	8
5.2.2	定义阶段	8
5.2.3	执行、测试阶段	8
5.2.4	审查阶段	8
6	敏捷开发调研	9
6.1	敏捷开发宣言	9
6.2	敏捷开发背后的原则	9
6.3	敏捷开发的概念	10

6.3.1	价值观、原则	10
6.3.2	以用户需求为核心	10
6.3.3	大项目划分为小项目	10
6.4	方法论-Scrum	10
6.4.1	Scrum 开发流程的三大角色	11
6.4.2	团队的 Scrum 实施过程	11
6.5	敏捷宣言的理解	12
6.5.1	以“人”为核心	12
6.5.2	研发工作的量化	12
6.5.3	为提升而重构	13
6.5.4	减负的艺术	13
7	处理需求变更	13
7.1	以“人”为核心的认知	13
7.2	贴近方法论的认知	14
7.2.1	预防为主，从需求源头做起	14
7.2.2	重视原型稿评审，重视需求确认	14
7.2.3	预留时间 BUFF	15
7.2.4	接不接受变更，按流程来	15
7.2.5	从四要素思考处理方案	15
7.2.6	变更研发交付方式	16
7.2.7	向客户反馈结果	16
7.2.8	按合同规定来	16
7.3	讨论结果	17
8	分工与总结	17

1 引言

1.1 日志编写目的

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。

《Team Work Log 2》整理罗列小队讨论当中，所涉及软件工程原理之间的逻辑，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。除了偏重方法论的软件工程原理，我们还在本文档中列举了我们基于团队项目 μ Vlogger

1.2 日志编写逻辑

软件工程中应当区分“价值观”与“方法论”。如果说团队理念是团队运作、理论付诸实践的基础，那么“流程”与“方法”才是实际上创造生产力的武器。本次实验关于生命周期模型、需求变更两大问题的探讨让我们队员深刻认识到了这一点。为此我们的报告将按照“理念”与“方法”两大方向展示小组讨论结果。

2 V 模型

2.1 V 模型定义

V 模型是一种传统软件开发模型，一般适用于一些传统信息系统应用的开发，而一些高性能高风险的系统、互联网软件，或一个系统难以被具体模块化的时候，就比较难做成 V 模型所需的各种构件，需要更强调迭代的开发模型或者敏捷开发模型。

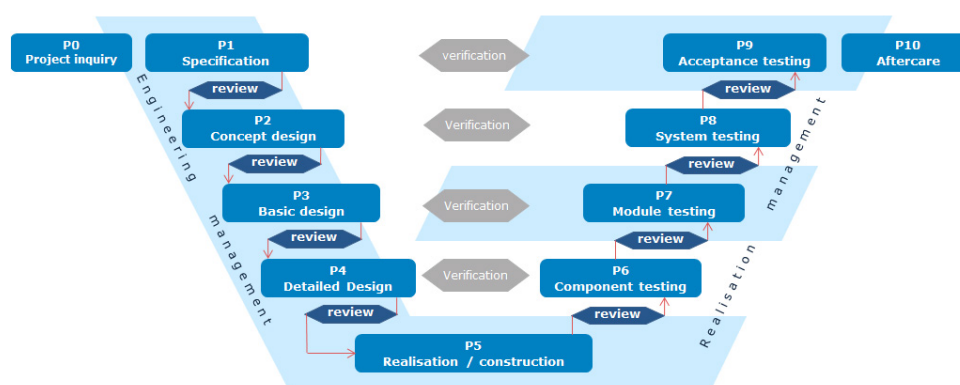


图 1: V Model Demo

RAD (Rap Application Development, 快速应用开发) 模型是软件开发过程中的一个重要模型，由于其模型构图形似字母 V (图 1)，所以又称软件开发的 V 模型。

它通过开发和测试同时进行的方式来缩短开发周期，提高开发效率。V 模型大体可以划分为以下几个不同的阶段步骤：需求分析、概要设计、详细设计、软件编码、单元测试、集成测试、系统测试、验收测试。对概要设计中表述的各模块进行深入分析，对各模块组合进行分析等，这一阶段要求达到伪代码级别，已经把程序的具体实现的功能，现象等描述出来。

2.2 V 模型流程

2.2.1 需求分析

即首先要明确客户需要的是什么，需要软件作成什么样子，需要有那几项功能，这一点上比较关键的是分析师和客户沟通时的理解能力与交互性。要求分析师能准确的把客户所需要达到的功能，实现方式，等表述出来，给出分析结果，写出需求规格说明书。

2.2.2 概要设计

主要是架构的实现，指搭建架构、表述各模块功能、模块接口连接和数据传递的实现等项事务。

2.2.3 详细设计

对概要设计中表述的各模块进行深入分析，对各模块组合进行分析等，这一阶段要求达到伪代码级别，已经把程序的具体实现的功能，现象等描述出来。其中需要包含数据库设计说明。

2.2.4 软件编码

按照详细设计好的模块功能表，编程人员编写出实际的代码。

2.2.5 单元测试

按照设定好的最小测试单元进行按单元测试，主要是测试程序代码，为的是确保各单元模块被正确的编译，单元的具体划分按不同的单位与不同的软件有不同，比如有具体到模块的测试，也有具体到类，函数的测试等。

2.2.6 集成测试

经过了单元测试后，将各单元组合成完整的体系，主要测试各模块间组合后的功能实现情况，以及模块接口连接的成功与否，数据传递的正确性等，其主要目的是检查软件单位之间的接口是否正确。根据集成测试计划，一边将模块或其他软件单位组合成系统，一边运行该系统，以分析所组成的系统是否正确，各组成部分是否合拍。

2.2.7 系统测试

经过了单元测试和集成测试以后，我们要把软件系统搭建起来，按照软件规格说明书中所要求，测试软件其性能功能等是否和用户需求相符合，在系统中运行是否存在漏洞，等。

2.3 验收测试

主要就是用户在拿到软件的时候，在使用现场，会根据前边所提到的需求，以及规格说明书来做相应测试，以确定软件达到符合效果的。

V 模型具有缩短开发周期和提高开发效率的优点，但同时 V 模型仅仅把测试过程作为在需求分析、系统设计及编码之后的一个阶段，忽视了测试对需求分析，系统设计的验证，需求的满足情况一直到后期的验收测试才被验证。要求研发人员和测试人员同时工作，从而尽快找出程序错误和需求偏离，减少成本。

3 顺序生命周期模型

通常用 V 型生命周期模型和瀑布生命周期模型来表示这种顺序的开发过程。而事实上，这两种生命周期模型有许多不同的形态，将不同的阶段引入生命周期模型，并在不同阶段之间设立边界。以下介绍的生命周期模型的各个阶段是经过许多最有经验的开发者经过反复实践而得来。

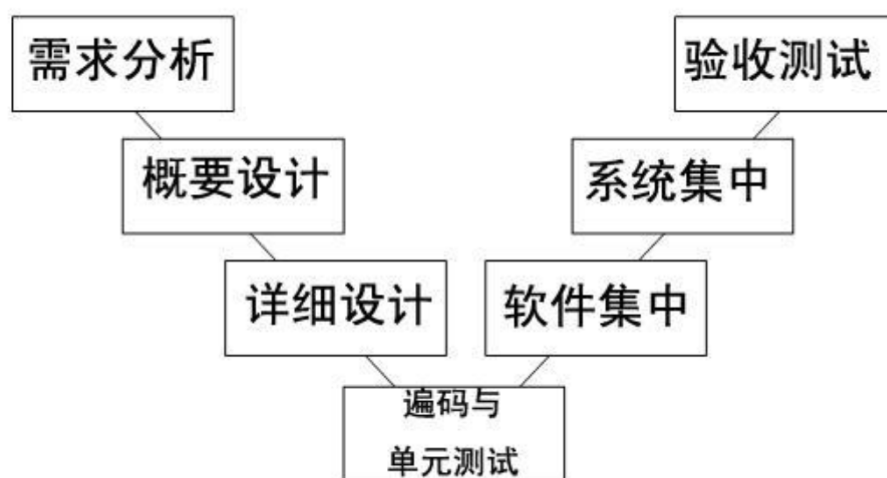


图 2: 顺序生命周期

4 渐进开发生命周期模型

软件开发所具有的一个问题就是用户急需软件产品，但是开发者却要花费很长的时间去完整地进行开发。那么取一个折中的解决方法就是节省一些时间，但在功能上打一

点折扣——开发出一个功能有所缩减的试用版给用户，作为完整版发布前的一个跳板。也可以将这个跳板作为软件减少软件开发风险的一种方式。

通常将这种方法称为渐进开发或是执行阶段。与之相对应的生命周期模型被称为渐进开发生命周期模型。在渐进开发生命周期之中，每一个独立的阶段都将遵从 V 型和瀑布型生命周期模型。

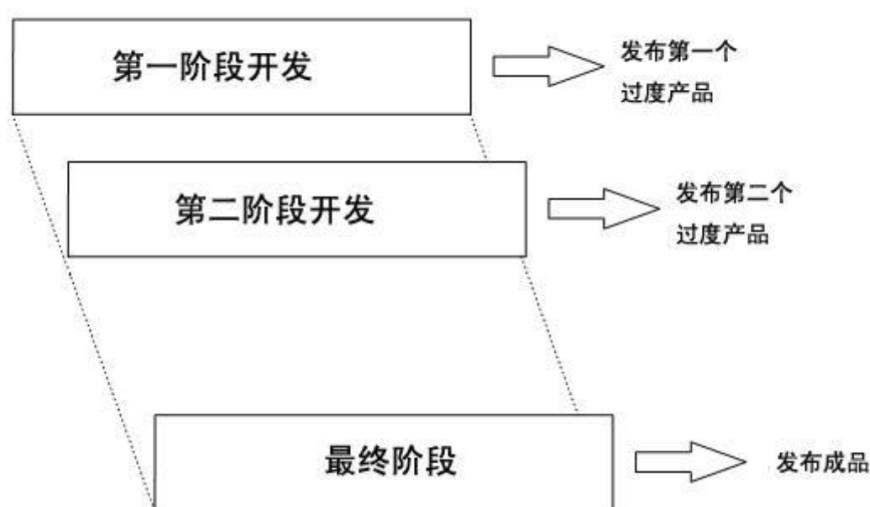


图 3: 渐进开发生命周期

5 迭代生命周期模型

5.1 模型定义

迭代生命周期模型并不是一开始就完全适应需求，而是先根据说明先开发软件的部分些可执行部件。而是先开发一些具有部分功能的部件，这些部件要求能够通过审查以确定更进一步的需求。不断重复这个过程，为此模型的每一个周期遍写出更新版本的软件。一个迭代周期模型由下图的四个连续部分组成不断重复组成。

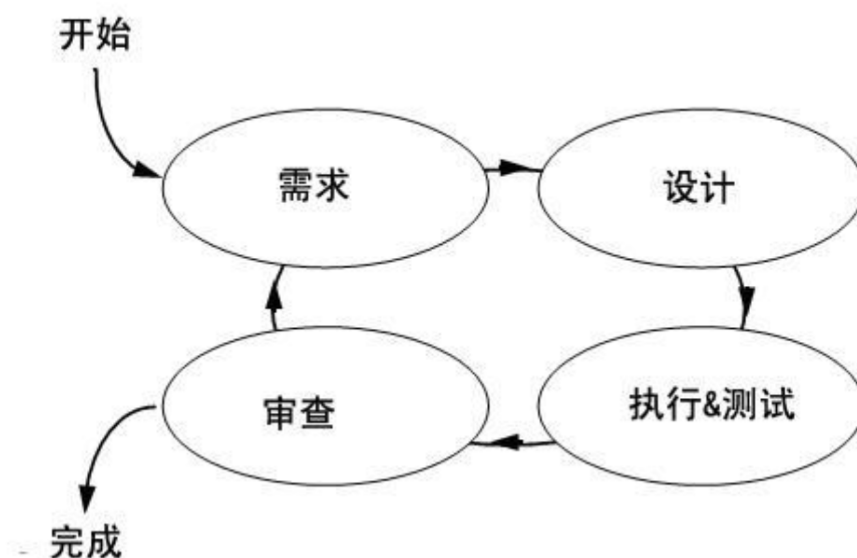


图 4: 迭代生命周期

5.2 迭代周期

5.2.1 需求分析阶段

在这个阶段，主要是收集并分析用户的需求，并且制定这个迭代模型最终并且完整的需求说明书。

5.2.2 定义阶段

在这个阶段，设计出适应需求的软件解决方案。这有可能是一个全新的设计，也有可能是原来设计的一个延伸。

5.2.3 执行、测试阶段

在这个阶段，将对软件进行编码，集中并进行测试。

5.2.4 审查阶段

在这个阶段，将对软件进行评估，对目前的需求进行审查，并对其进行修改和更新。

在迭代模型的每一个周期，都要作出一个决定：要将编写出的软件抛弃，还是作为下一个周期的起点。如果软件完全符合了需求，那么就可以进行发布，否则就是一个失败的开始。

迭代模型的前三个阶段就好比是简化版本的 V 型模型或是瀑布模型。迭代模型中的每一个周期所编写出来的软件都要为软件的集中，系统集中和验收进行单元测试。在迭代模型中软件的开发经历了多少个这样的周期，那么就要进行多少次这样的测试。

6 敏捷开发调研

6.1 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.

6.2 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.

6.3 敏捷开发的概念

6.3.1 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。

6.3.2 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。

6.3.3 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。

6.4 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。

6.4.1 **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和一定的表达能力；成员可以采用任何工作方式，只要能达到 **Sprint** 的目标。

6.4.2 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：如果再開回顧會的時候，有人提出需求不清，拆解不夠細緻，工時估算不准之類的问题，那你真是太幸運了。碰到這種情况，你順勢提出這個制度：每個版本開發前，產品經理要和團隊人員開個會，一起逐條對需求。產品經理要向團隊解釋每個需求，並且對團隊提出的疑問進行解釋和澄清。開發團隊要將需求分解成任務，估算工時，最終形成一個開發清單。
5. 引入評審會議：當計劃會成功開過幾次之後，就可以考慮引入評審會了。你可以在某個重大功能發布的時候，把所有項目相關的人員召集起來，向大家展示團隊做了哪些了不起的功能，也可以和大家討論討論還可以做哪些優化。
6. 引入 Scrum：當整個流程都跑順以後，就可以正式引入 Scrum。此時需要任命 PO、SM，對團隊培訓 Scrum 知識，講解 Scrum 的價值觀。

6.5 敏捷宣言的理解

6.5.1 以“人”為核心

章節 6.1 當中值得強調的是，作者並沒有否認 **right value**，而是側重於 **left value**。也就是說，儘管比其文檔而言，敏捷開發看重實際代碼，但是：文檔一樣重要。敏捷開發的轉移對傳統工程中繁重部分的注意力，而致力於確實推進軟件項目的部分。

敏捷宣言原則與傳統軟件開發方法很明顯的不同在於，就是讓我深刻感受到了工程項目中的“人”的元素。軟件開發是“人與人的合作”，軟件開發的目的也是滿足“人的需求”（章節 6.2 中倡導“欣然接受需求變更”）。一切都是為了活生生的人服務，所以我們在敏捷開發當中促進開發者的有效溝通交流、尽可能地滿足人的需求。

6.5.2 研發工作的量化

研發工作往往是比较难以量化的，對於研發管理來說，即使把迭代周期壓縮到了 1-2 周，對於進度的控制粒度也是遠遠不夠的。我一般会要求團隊從第一天開始就尽可能的實現持續集成（每日構建），哪怕產品啥都沒有，不僅僅是工程團隊，產品所有相關的人從第一天開始就可以真實的運行產品，這比什么都更直觀的反應了產品進度到哪裡。

在持續集成中還有很多相關的規範需要團隊執行，比如自動化測試，對於一些核心代碼尤為重要，有了自動化測試的用例覆蓋，對於每天構建的版本質量就有了更清晰和量化的數據。自動構建的结果會每天發送郵件的團隊所有人，每個人都可以清楚的了解到項目的進展情况。

6.5.3 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。

6.5.4 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。

7 处理需求变更

7.1 以“人”为核心的认知

敏捷开发发掘“人”的元素，尽管合乎情，但是个人理解当中，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。

7.2 贴近方法论的认知

作为项目经理，应对需求变更的第一要点，就是让整个团队都知道，需求变更是经常会发生的事情，是再正常不过的事情。发生需求变更，项目经理需要引导团队思考两个问题：

这个需求变更该如何处理？

下次如何预防这类需求变更不再发生？

7.2.1 预防为主，从需求源头做起

客户口中的需求，经常会站在他们的立场上提出来，而且经常提需求的时候也提解决方案。换句话说，他们对于什么是需求、什么是解决方案分不清楚。

在实际操作中，针对不同类型的项目，项目经理可以有如下的应对：

对于实施类项目，在同时存在项目经理和产品经理的团队中，项目经理不要把需求沟通的工作全权交给产品经理，项目经理也必须参与需求收集、理解需求、讨论需求。

尽量获得一手需求，站在项目落地风险的角度，帮助产品经理一起识别客户的伪需求和不合理的需求。

要有质疑的勇气，否则等到错误的需求流转 to 交付团队，到时变更成本更高。

对于产品类项目，需求的相关工作以产品经理为主导，项目经理可以少操些心，但是当交付过程中发现需求频繁变更时，项目经理就得多留意需求分析阶段产品经理的产出物了。

7.2.2 重视原型稿评审，重视需求确认

对于实施类项目，把做好的原型稿或视觉稿尽早展示给客户，并且和客户进行需求确认，邮件、微信、签字等方式都行，形式不限。

对于产品类项目，重视原型稿和视觉稿评审。

要让问题在项目早期暴露出来，问题发现得越晚，处理成本越高。

7.2.3 预留时间 **BUFF**

既然需求变更是不可避免的事实，那么，项目经理在排期时就需要留有一定的时间 **BUFF**。

7.2.4 接不接受变更，按流程来

实施类项目，需要有一个是否接受客户需求变更的流程。

收到需求变更后，项目经理组织团队先评估需求变更将产生的新成本，按照项目的具体情况决策是否接受变更（由项目核心团队或决策委员会 **CCB** 决策）。

产品类项目，更多的矛盾点集中在：要做的功能太多，资源太少。这时候制定合适的优先级规则就尤为重要了。

如果产品的功能不多，团队规模较小，一般就是产品经理、项目经理、研发和测试的负责人组成的核心团队做决策就好了。

如果功能点几百个，涉及的团队也是大几十号人，这时候就需要对优先级的规则进行量化了，可以从这几方面入手：

1. 带来的新收入
2. 影响的用户数
3. 提升用户的活跃度
4. 投入的人力成本
5. 投入的时间成本
6. 功能的权重

有了功能优先级之后，再根据现有功能实现的具体进度，挑选哪些功能往后放，新出现的功能插到哪个版本里面去。

7.2.5 从四要素思考处理方案

如果团队决定接受这个需求，项目经理接下来就要看团队怎么处理这个需求了。

项目经理可以从项目管理四要素出发，看下能改变哪一个要素：

1. 是降低系统的质量要求？ - 比如留些 bug 以后处理
2. 是改变项目的交付范围？ - 比如多了一个功能进来总得砍一个出去
3. 是延长项目的交付时间？
4. 还是改变实现功能的方式？

给出合理的需求处理解决方案，是项目经理综合能力的体现。把变更的后续跟进方案通知到整个团队，并且存档变更纪录。

7.2.6 变更研发交付方式

如果项目的需求就是要经常变更，还有一种常见的办法就是把瀑布流的研发交付方式改成敏捷交付或迭代交付，小步跑，多优化，主动拥抱需求的变更。

7.2.7 向客户反馈结果

实施类项目，无论团队内部讨论的结论是接受还是不接受变更，都要把跟进方案与客户进行沟通。

找到客户中懂软件、懂项目管理、能做决定的人进行沟通。

找到对的人后，可以带着准备好的几套跟进方案与其沟通，让他们明白需求变更给项目带来的影响：团队是投入了额外的人力，还是去掉其他需求，还是需要延期交付，这些变化都需要让客户知道。

另外，如果变更涉及到项目成本的变化，需要交涉合同金额，项目经理最好拉着商务一起去沟通。

7.2.8 按合同规定来

实施类项目，需求会体现在合同的工作说明书 (SOW) 中。

所以，为什么前面说，即使团队中有产品经理，项目经理也务必要参与到需求沟通中，是为了特别留意客户的需求在 SOW 中会不会漏掉、会不会表述不清。

既然需求都体现在 SOW 中，当客户提出新需求时，项目经理可以以合同中没有规定为由拒绝

7.3 讨论结果

章节 7.1 以及章节 7.2 的探讨，展示了我们小组将问题由主观认知到方法论认知的过渡。最终我们达成了一致的理解。

概括而言：心理上欣然接受需求变更，方法上量化及妥善处理变更。

8 分工与总结

本周组长轮换，由曹远担任组长。

人员	调研工作	撰写工作	工程工作	组内活动
曹远	V 模型	开发计划、测试计划初步	测试工具评估, 组内报告	项目经理、组织讨论
苑宗鹤	多种生命周期模型变式、需求变更	开发计划、小组日志、需求分析	架构设计、后端初步框架	后端负责人、负责引入站会制度
张火亘	瀑布模型	开发计划、跟进前端 UE、UI 设计	前端基础框架搭建	前端负责人、引入任务可视化
高德琛	敏捷开发理念及 Scrum 方法、需求变更	开发计划、小组日志、测试计划	数据集预处理、模型设计	算法负责人、Scrum 准备