

04 | 敏捷之道：大型Go项目的开发流程是怎样的？

2022-10-18 郑建勋 来自北京

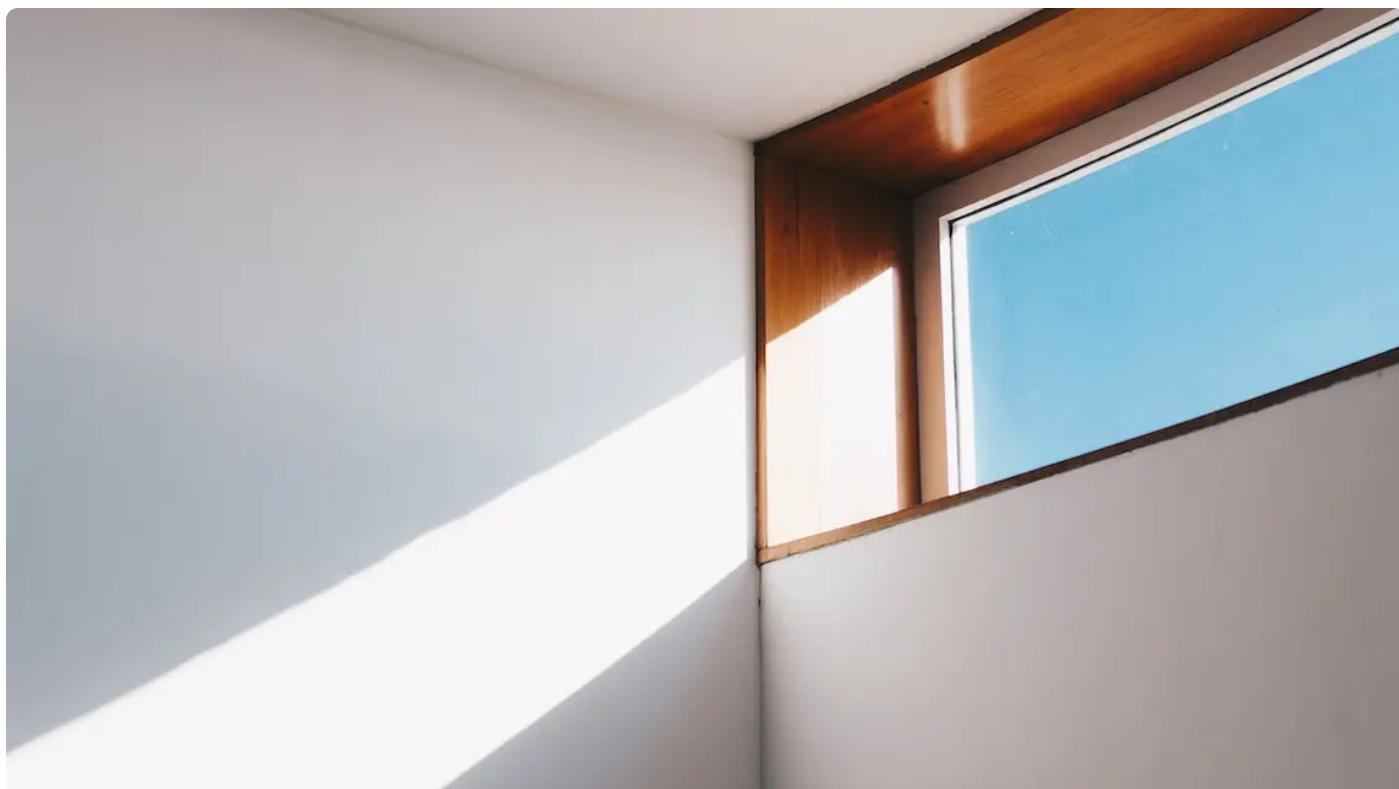


天下无鱼

<https://shikey.com/>

《Go进阶·分布式爬虫实战》

[课程介绍 >](#)



讲述：郑建勋

时长 13:19 大小 12.17M



你好，我是郑建勋。

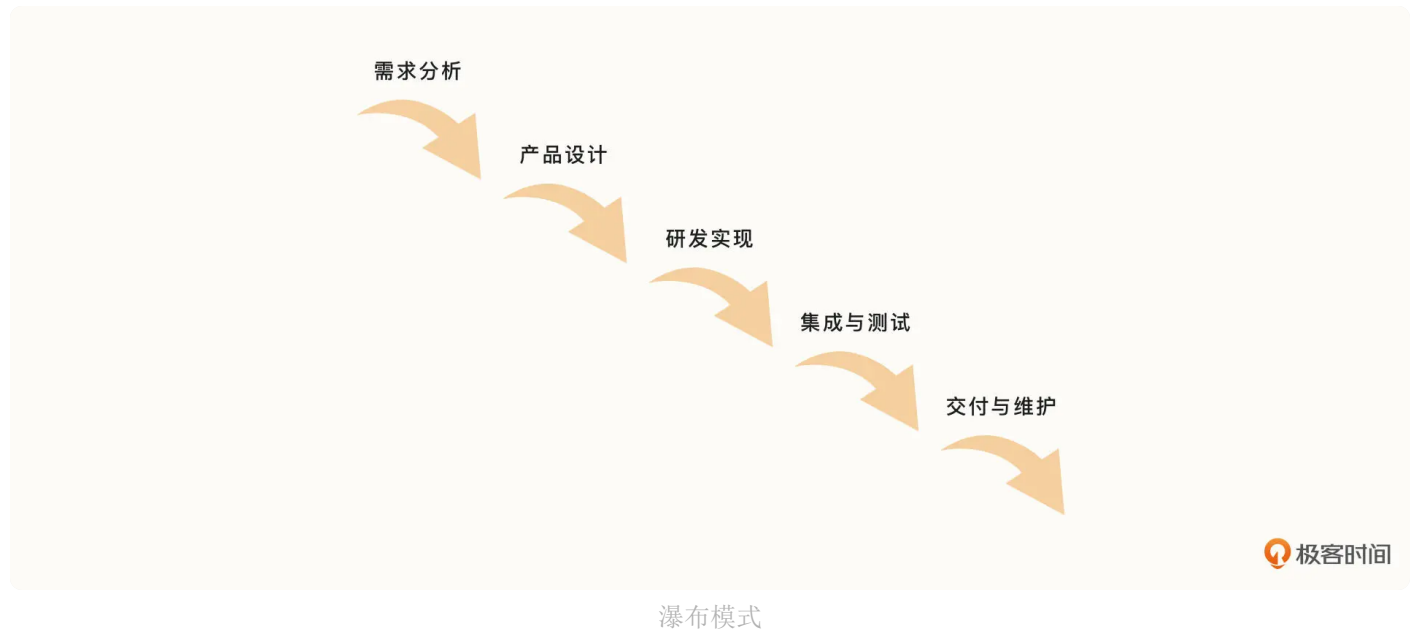
前面，我们介绍了和 Go 语言相关的基础知识与学习方法，但仅仅掌握高级语言的语法与原理还不足以让我们完成一个真实的项目。

要想完成一个项目，需要遵循一些基本的开发流程。一个优秀的开发流程可以帮助我们识别和降低开发过程中可能面临的风险，助推项目的开发进度。

但是高效研发流程的最佳实践是怎样的？针对具体的项目，又应该选择怎样的研发流程？接下来两节课，我将为你展示大型互联网产品的整个开发流程，帮助你理解复杂流程背后的设计、技术和工具，方便你搭建起适合自己团队的开发流程。

瀑布模式 vs 敏捷模式

谈到项目的开发流程，有过软件开发经验的同学脑海中大致会出现市场调研、需求分析、产品设计、研发实现、集成与测试、项目交付与维护这些阶段。传统的研发流程确实是这样连续的过程，只有完成前一个阶段的验收审核才能够开始下一个阶段，这种开发模式也被称为瀑布模式（Waterfall Model）。瀑布模式借鉴了传统工业生产的预设性方法，它也是大多数软件开发的最初标准。



瀑布模式比较适用于下面的场景。

- 需求在规划和设计阶段就已经确定了，而且在项目开发周期内，需求没有或极少有变化。例如航空航天或者金融核心系统等。
- 团队对这一技术领域很熟悉，风险低、规模小。
- 合同式的合作方式。项目的开发严格依据说明，客户需求明确且不参与软件实现过程。

打个比方，一家人工智能公司的人脸识别技术已经相对成熟了。这家公司需要为某大型体育馆开发一套识别观众是否佩戴口罩的系统。在签好合同之后，需求就基本固定了，不存在特别大的变更。客户接下来只要等着在指定日期验收项目就可以了。而且，这种需求只涉及对应人脸的模型训练和上层系统的适配。团队对这种技术比较熟悉，风险很低，这种情况就比较适合瀑布模式。

但是，瀑布模式也存在很多问题。由于用户只能在产品交付后才能提供有价值的反馈。因此，如果需求被遗漏或者一开始就是错误的，在开发后期变更需求就可能付出很大的代价，导致成本或时间超支。

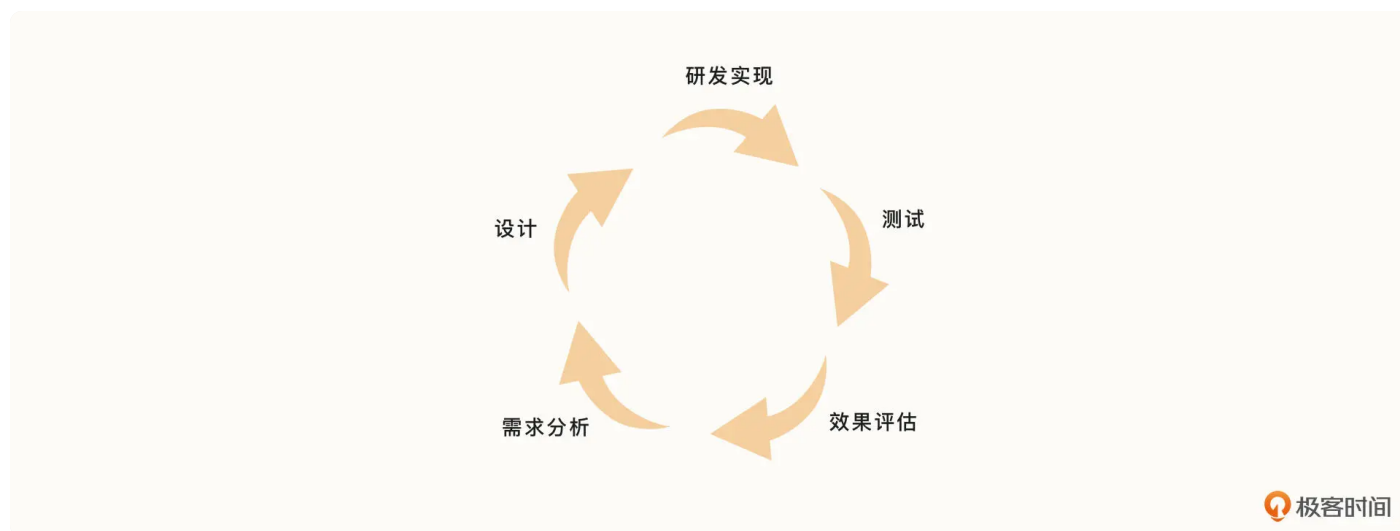
也正是因为，瀑布模式已不再是当前主流的开发模式，另一种更流行的开发模式叫做敏捷开发（Agile development）。



敏捷开发描述了一套软件开发的价值和原则，如果你感兴趣的话可以看看 [敏捷宣言](https://shikey.com/) 提出的 4 种价值观和 12 条原则。敏捷宣言是 2001 年由数十位行业专家达成一致的敏捷软件开发方法。

敏捷开发的核心思想是拥抱变化，强调对于变化的适应性，更强调开发者与业务专家、客户之间的互动，强调持续改进和持续交互产品，持续提高客户满意度。

和多阶段顺序开发的瀑布方法不同，敏捷模式强调的是迭代：



敏捷模式强调迭代

敏捷开发是增量构建的，每次迭代都满足总需求的一部分，而不是试图一次性交付所有功能。这使新功能的价值可以得到快速验证。

有多种实现了敏捷开发思想的框架供我们使用，比较知名的 Scrum、看板（Kanban）、极限编程（XP）、精益软件开发（Lean Software Development）等。

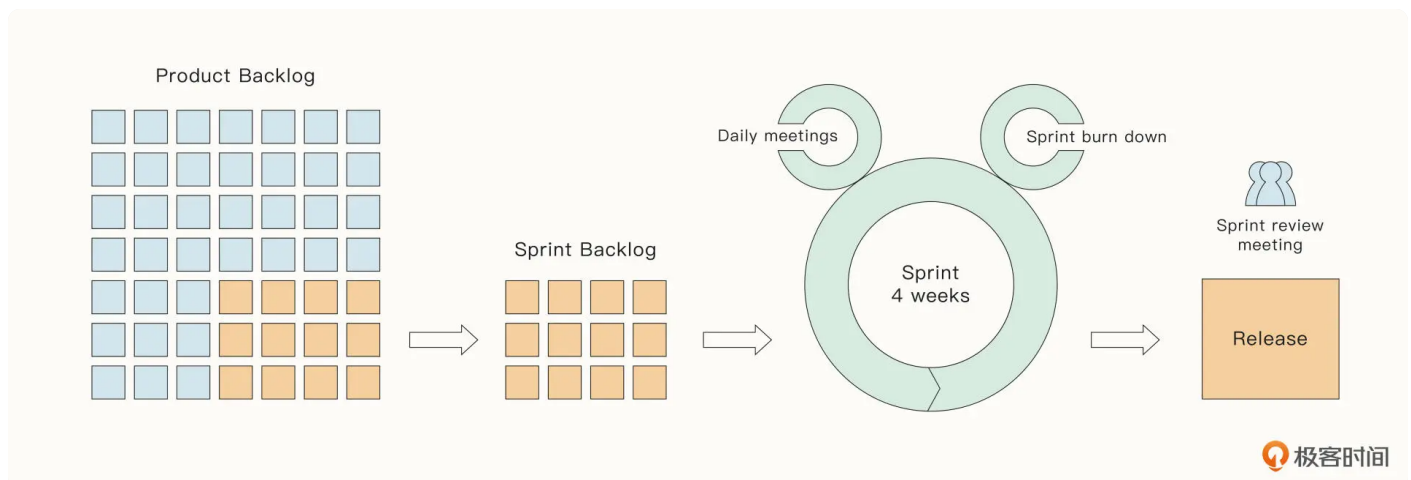
以当前最流行的敏捷框架 Scrum 为例，它包括下面几个重要的阶段。

1. 产品经理基于项目的愿景（Vision），收集产品待办事项清单（Product Backlog）并确定优先级。
2. 团队成员根据阶段性的成果将项目阶段拆分为一个个的 Sprint，即冲刺周期或开发周期。每一个 Sprint 开始的时候，需要开一个 Sprint 会议，把产品待办事项清单里的事项添加到当

前 Sprint 中。添加的原则是，需要考虑 Sprint 的交付价值以及事项的优先级。当前 Sprint 清单里面的待办事项也被称为 Sprint Backlog。



3. Sprint Backlog 可以由团队成员拆分，并细化为每一个成员每天具体的工作任务。成员们可以根据任务进度去看板上更新任务的状态。
4. 在当前 Sprint 运行时，团队成员要参加每日站立会议（Daily Scrum Meeting），每次会议时间控制在 15 分钟内。会上成员要根据看板的内容逐个进行发言，向所有成员汇报昨天已完成、今天待完成的事项，交流不能解决的问题。会议结束后，要及时更新项目的燃尽图（Burn Down Chart），以便跟踪项目进度。
5. Sprint 结束后，团队成员一起评审（Sprint Review）本次 Sprint 的产出。这个产出物（release）可能是一个可以运行的软件，也可能是一个可展示的功能。每个人都可以自由发表看法，协助产品负责人对未来工作做出最终决定。并根据实际情况，适度调整产品待办事项列表。
6. Sprint 结束后，大家聚在一起开一次回顾会（Sprint Retrospective），回顾一下团队在流程和沟通等方面的成效。一起讨论哪里完成得好，哪里还需改进。



大型互联网产品的研发流程

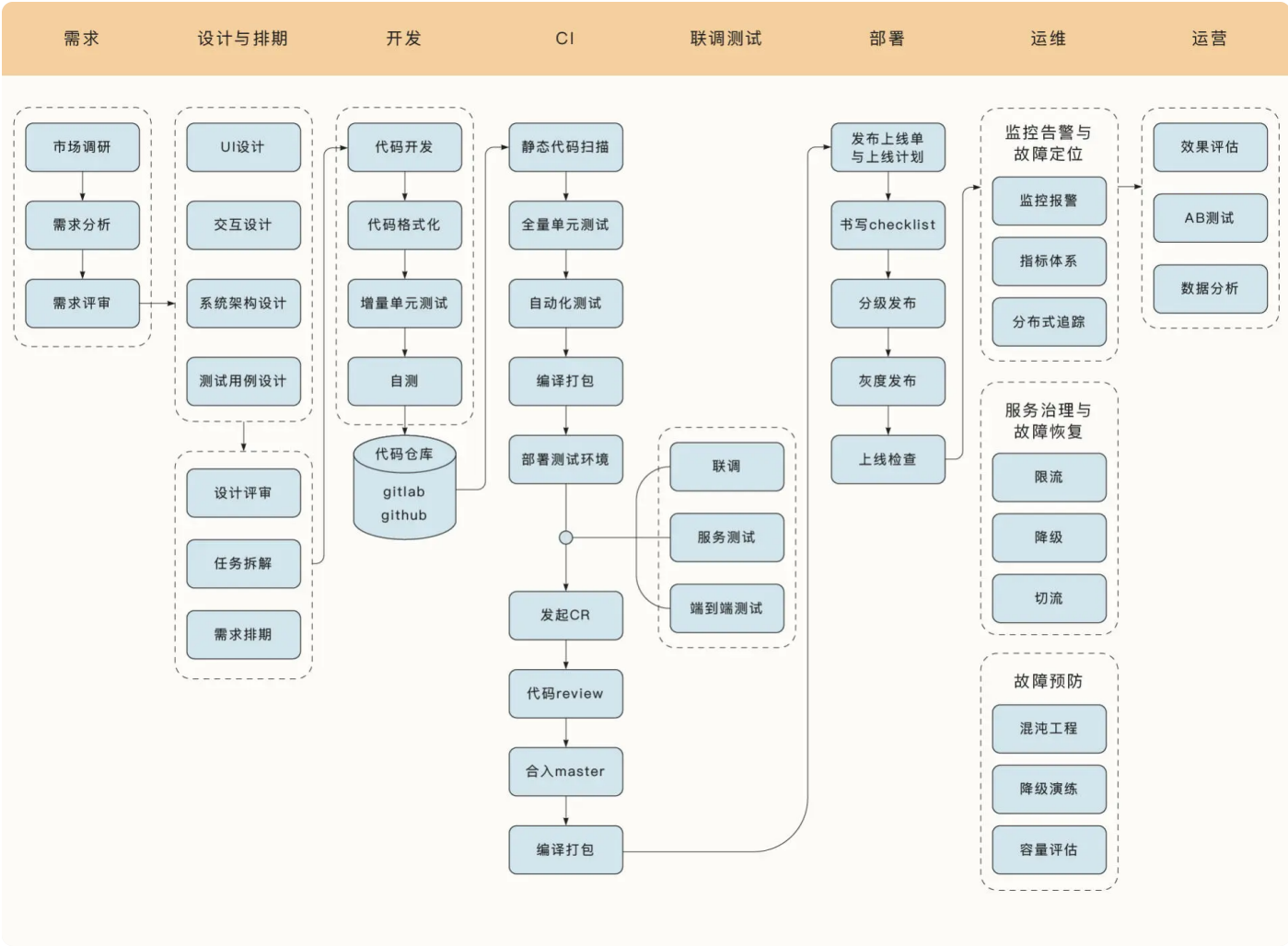
通过上面的讲解可以看到，瀑布模式和敏捷模式这两种开发模式主要是研发形式上的不同，一种是顺序开发，另一种是迭代开发。但从需求分析到项目上线交付的过程，瀑布模式和敏捷模式具体要做的内容是类似的。

例如我们都需要需求的调研、项目的开发与测试。由于公司业务、所处阶段或者管理者理解的不同，每家公司对研发流程的设计都有较大差异。

例如，需求分析阶段可能涉及到如何进行需求调研、需求评审，如何编写需求文档等问题。但对一个小公司来讲，团队规模小，目标比较集中，缺失需求评审等步骤可能不会带来多大的问题。不过，随着公司规模越来越大，为了方便对项目、流程、人员进行管理，这时候通常会更加注重项目流程的规范、项目进度和人员时间的管理，从而规避项目风险、保证项目质量和按期交付。

那么问题来了，具体到每家公司应该如何选择开发流程呢？

我们可以先来看一看大型互联网公司的研发流程。大型互联网公司的研发流程全面、严谨，且经过了时间的验证，有助于其他公司借鉴、查漏补缺，以此设计适合自己的开发流程。大型互联网公司典型的开发流程如下所示：



接下来，让我们具体来看看大型互联网公司开发流程中的各个阶段。

需求阶段

任何项目的第一步都涉及到需求的收集、识别与需求的分析。这样我们才能确定有哪些项目可以做并且值得做。



正所谓方向不对，努力白费。需求的重要性再怎么强调都不为过。正确的需求识别与需求分析对于成功的项目至关重要，因为它会影响所有后续阶段，**而缺陷在软件流程中停留的时间越长，它对下游造成的损害就越大**。需求可以分为三种类型：商业需求、功能需求和非功能需求。

1. 商业需求

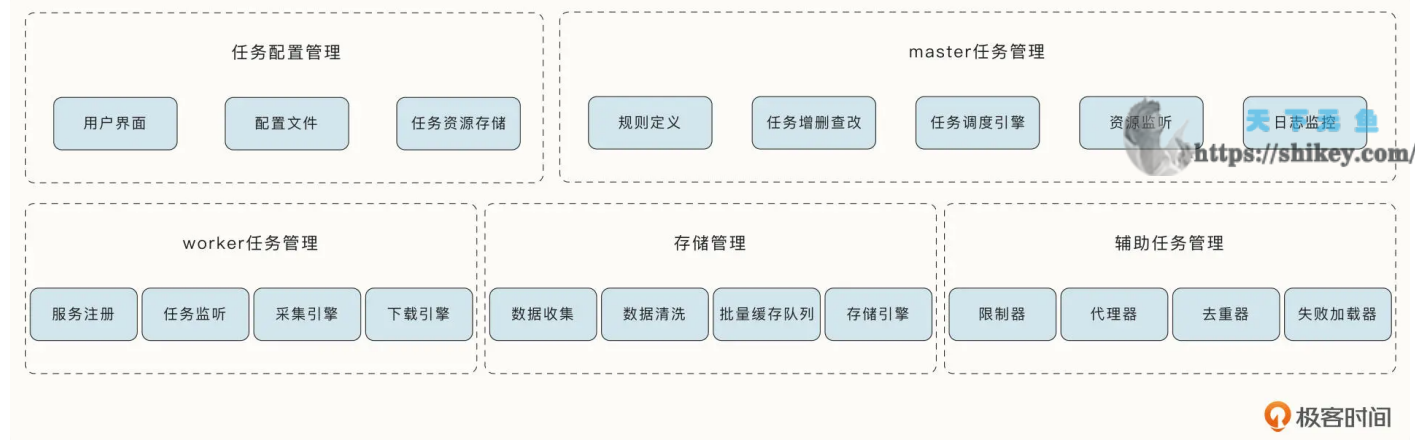
商业需求定义了软件发现的商业机会或者要解决的商业问题。商业需求可能来自于市场调研、数据分析、竞争对手分析等。例如，项目能提供竞争对手没有提供的功能，或提供某种经过改进的功能（如更快的响应时间）来获得竞争力。

商业需求通常会影响软件系统的质量属性。在需求分析的最初阶段需要出一份**市场需求文件（MRD, Market Requirements Document）**，主要讨论项目的商业价值，用于决策是不是真的要做这个项目。需要讨论的话题包括：

- 这是一个什么样的产品？
- 我们的目标客户是谁，解决了什么痛点？
- 这个产品在市面上有哪些竞争商品？
- 为什么要做这个产品，产品的销售策略、收益与风险是什么？

2. 功能需求

功能需求描述了软件的功能。例如，一个购物商城需要提供商品展示功能、购物车功能、支付功能。又如我们开发的爬虫项目，需要具备任务的增删查改、任务调度、代理、限流等功能。



功能需求详细描述了软件系统在行为方面的能力，开发者后续需要完成对应功能的开发。许多项目不成功的主要原因就是不充分的用户调研、不完整的功能需求或者误解了功能需求。

3. 非功能需求

非功能性需求，是指软件产品为满足用户业务需求而必须具有的，除了功能需求以外的特性。包括可维护性、可用性、可移植性、可测试性等。

举个例子，系统可用性指的是一个系统处在可用工作状态的时间的比例。如果一个系统出现崩溃等问题（例如不能点外卖，不能打车），这个状态就被叫做“不可用”。我们通常会用 N 个 9 来衡量服务的可用性等级。例如 3 个 9 代表服务在 99.9% 的时间内是可用的，转换为时间即年度允许的累积宕机时间为 8.76 小时。4 个 9 代表服务在 99.99% 的时间内是可用的，转换为时间即年度允许的累积宕机时间为 52.6 分钟。

识别出上面三类核心需求，明确了要实施该项目后，就需要对这些需求进行分析了。对于敏捷开发来说，当涉足到一个新的领域通常会实施一个最小可行性产品（MVP，Minimum Viable Product）。最小可行性产品只关注最核心的价值交付，即快速交付最基本功能的产品，以便快速验证产品的效果，快速抓住市场。

需求分析需要细化需求，而且还要确保所有利益相关者都理解它们。相关人员要仔细检查这些需求，以此确认是否存在错误、遗漏或其他缺陷。

需求分析包括将需求拆分、构建原型、评估需求的可行性和优先级、识别风险与约束。这一步需要产品经理准备 PRD（Product Requirements Document），PRD 包括项目背景、产品结构、功能流程、原型图、需求说明等，并拉上利益相关方（设计师、测试经理、架构师、研发工程师）进行需求的评审。

设计阶段

明确需求之后，技术人员需要根据需求完成方案的设计，从而确定项目的具体排期。相关人员的分工包括：

- **UI 设计师（User Interface Designer）**与产品经理沟通视觉细节。如果把项目开发比作是建造房屋，UI 设计师则要对房屋的外观进行设计。UI 设计师要设计产品的颜色、尺寸、形状等要素，输出界面效果图。
- **交互设计师（User Experience Designer）**了解用户的思维方式和行为习惯，设计交互流程界面，让产品更容易被用户使用，提供完美的用户体验。例如，一次打车服务涉及到从预估价格、使用优惠券、派单、等待接驾、查看行程信息、修改目的地、查看订单信息、结束行程等诸多环节，每一环都可能影响用户的体验。
- **系统架构师**需要对系统架构进行设计，对技术进行选型。例如，如何拆分微服务、如何保证分布式系统的一致性、选择哪一种开发语言、框架、中间件和数据库等。
- **研发工程师**需要设计技术方案，梳理功能流程，明确接口定义和上下游的调用流程，选择合适的算法与数据结构以保证程序的效率目标。
- **测试工程师**为了验证某个需求是否实现、是否存在缺陷，在测试之前会设计一套详细的测试方案和测试集。测试集某种意义上也定义了我们要实现的功能，测试合格意味着系统功能满足需求。

明确设计方案之后，就要对需求进行排期了，包括确定好开发时间、联调时间、QA 测试时间、以及上线时间等。

大公司一般都会有专门的项目管理工具方便我们管理整个项目的进度，包括了责任人、重要节点的截止日期、当前任务的状态等重要信息。



知名的项目管理工具包括用于 issue 跟踪和敏捷项目管理的 Jira、腾讯敏捷产品研发平台 TAPD、阿里的团队协作工具 Teambition 等。

总结

这节课就先讲到这里，我们总结一下。



好的研发流程能够高效管理项目资源、识别和降低开发过程中面临的风险，加速项目的研发进度。

研发流程的最佳实践包括需求分析阶段、设计阶段、开发阶段、测试、上线和运维阶段。当今的大型项目更多地实践了敏捷开发的思想，能够快速响应新的需求，体现出了对变化高度的包容。

不过在研发流程的每一个阶段，不同项目又可谓别有洞天。我们这次课程介绍了大型互联网产品的研发流程细节，其背后体现了大量的设计智慧，值得参考借鉴。

在本节课我们介绍了项目开发的需求和设计阶段，在下节课，我还会继续介绍大型互联网产品开发流程中关于开发、测试、上线以及运维阶段需要遵守的流程与规范。


课后题

最后，我也给你留两道思考题。

1. 有些人觉得 QA 测试并不是必要的，因为开发者更了解项目的细节，也理应做到充分的测试。你觉得呢？
2. 在课程中我介绍了 Scrum 敏捷框架的流程和优势，你觉得 Scrum 框架的缺点在哪里，有哪些项目并不适合使用 Scrum 框架？

欢迎你在留言区与我交流讨论，我们下节课再见！

分享给需要的人，Ta购买本课程，你将得 20 元


 生成海报并分享

 赞 3  提建议

精选留言 (6)

 写留言



抱紧我的小鲤鱼 

2022-10-18 来自江苏

qa测试是必须的，虽然开发对项目很熟，但会思维固化，不容易发现逻辑之外的错误，当局者迷 旁观者清。以前公司试图让开发做交叉测试，去掉qa，但实际效果可以说几乎没有



 3



Realm

2022-10-18 来自浙江

1 QA测试很有必要，不能保证研发对每个功能都做了充分的测试，研发有时候对用户输入、边界条件考虑不妥，这些可能在QA阶段可以测试出来；

2 敏捷开发对需求固定，变更较小的项目必要性不大，会觉得“形式大于内容”；



 2



Howe

2022-10-18 来自北京

思考题：

1、QA 测试应该是必须存在的一个环节，虽然开发者是应该做好测试，但是开发者也只能做好自己负责模块的测试，特别是在微服务环境下，每个开发者只关注自己负责的服务，对于系统整体的集成测试、性能测试，还是需要专门的 QA 来完成。另一方面，QA 其实也可以对开发者所交付的东西的质量进行监督，可以发现开发者疏忽的问题。

2、Scrum 框架的缺点：感觉 Scrum 框架更讲究迅速，看起来更适合小型、要求先快速交付一版的新项目，很多环节由文档转变为面对面沟通，对于长期迭代的项目来说，可能会导致一些重要材料的丢失，如果项目人员流动大，可能会对后续的长期维护埋坑。

不知道是否理解正确。

另外也有一个问题想请教老师：以上内容讲的是一个软件开发生命周期，即便是在网上查找资料，也都是类似的内容，但是有 2 个疑惑：

1、对于长期维护和迭代的项目来说，可能一个项目做了好几年了，甚至运行了十年了，那么每一个功能需求都需要走全部流程吗，包括需求分析、设计方案的制定等等？

2、对于一个运行很长时间的复杂项目来说，在设计方案的时候怎么去避免冗余、重复等问题，举个例子，假设项目中某个子模块存储了用户信息数据，现在来一个新需求也要这些数据，或者说，一个功能里写了一个工具包，另一个新需求也需要这方面的功能，开发者怎么

在设计方案的时候就知道原本项目里已有这样的功能，从而直接使用就好，不用重复开发？
当一个项目很大时，可能架构师也未必能对整个项目的方方面面都了如指掌，即便可以耐心地对方案进行细抠，但是这可能需要花费较多时间，又会影响需求完成的进展。[天下无鱼 https://shikey.com/](https://shikey.com/)
不知道对于上面的问题，老师有没有什么经验可以分享下，谢谢老师~

作者回复: 第一个问题要根据实际情况来看，就算是很老的项目，但是人员却是不断在流动的，如果不留下资料，后面的人两眼一抹黑。另外越老的项目越容易犯错，因为如果前期没有经过设计和遵守规范，随着业务越来越复杂，牵一发就都能全身了，所以设计方案并让团队成员review是有必要的，至于需求分析更有必要了，通常是产品经理给出的，决定了为什么要做这个项目以及要做什么事情。

第2个问题，主要还是项目的组织不合理，这是实践中常常遇到的问题，功能都混杂在一起了，比如一个用户的鉴权功能不能每一个功能模块都写一份自己的吧。合理的抽象，把对应的功能模块能够清晰的抽离出来，还有基础的工具库，通用的能力单独出来，我在后面设计项目架构的时候还会详细说。



thinkings

2022-10-19 来自北京

请问老师课程中的课件，是否可以提供下载

作者回复: 你指的课件是什么呢，PPT？

共 2 条评论 >



一步

2022-10-19 来自广东

Scrum 最大的问题就是会议太多了，最后会流于形式。 要根据实际的情况进行改进



东

2022-10-18 来自上海

M

