

## 设计篇

# 从业务诊断到形成方案

从本篇开始，我们将正式进入 B 端产品设计的主题：如何从零开始构建一套 B 端产品，来支持一条业务线。这其实是相当有挑战的，设计人员要完成业务梳理、流程设计、组织架构设计、数据建模、界面设计、权限设计等一系列工作，既要有对宏观的把控能力，又要有对细节的专注力。

为了便于大家理解，我们将结合一个完整的案例——设计一个分销平台，来讲述 B 端产品设计中各个环节的方法、要素和技巧。各章的主要内容如下。

第 3 章概述 B 端产品设计的一般流程，并引出贯穿全书的 M 公司案例的背景；第 4 章介绍 B 端产品的业务调研，这是产品设计开始之前的重要工作；第 5 章介绍 B 端产品的整体方案设计，一起分析并勾勒出产品的轮廓；第 6 章介绍 B 端产品的细节方案设计，包含 B 端产品细节设计的所有关键主题，例如业务建模、界面设计、数据埋点等。第 7 章介绍 B 端产品经理应该具备的技术常识，理解这些技术常识，对日常工作会有很大帮助。

很多产品经理入行后，可能一直在做迭代优化的工作，没有机会参与从 0 到 1 设计一套产品的过程。当积累了足够多的经验后，产品人一定要为自己争取或创造机会，亲历一次从零开始的产品设计与落地的过程，锻炼自己的全局性思维和设计能力。

现在，让我们做好准备，开始这场有趣的 B 端产品建设旅程吧！

## 第 03 章

# B 端产品建设概述

B 端产品往往涉及复杂的业务关系和场景，该如何设计并实施一套 B 端产品呢？其实是有规律可循的，遵循标准的流程逐步开展工作，可以提升效率、少走弯路。本章将从总体上介绍 B 端产品建设的一般流程，以及流程中每个环节的要点。此外，我们还将对比 B 端和 C 端产品建设流程的区别，帮助大家理解二者的不同。

## 3.1 B 端产品的总体建设流程

开展 B 端产品建设时，会面临两种可能的业务情况：

- 业务还未开展，只讨论了初步的可行性，需要设计最低成本的试错方案。
- 业务已经通过线下的初步验证，现在需要系统支持，实现线上化，全面推进业务。

第一种情况不需要设计完整的产品，只需要设计一个方案，让业务以最低成本做初步尝试，论证可行后再考虑产品化支持。

第二种情况需要做全面的产品化支持工作，我们要讲的就是这种情况下的总体建设流程。

B 端产品的总体建设流程需要借鉴软件工程自顶向下的设计思路，从抽象到具体逐步展开工作，大体上可分为业务问题诊断、设计解决方案（包括整体方案和细节方案）、执行并优化解决方案（又分为设计技术方案、实施、迭代）三大阶段，每个阶段包含具体的关键步骤，不同阶段会涉及不同的参与人员，如图 3-1 所示（其中 PM 代表产品经理）。按照这个流程和思路来梳理业务、进行产品设计，比较容易保障工作效果。

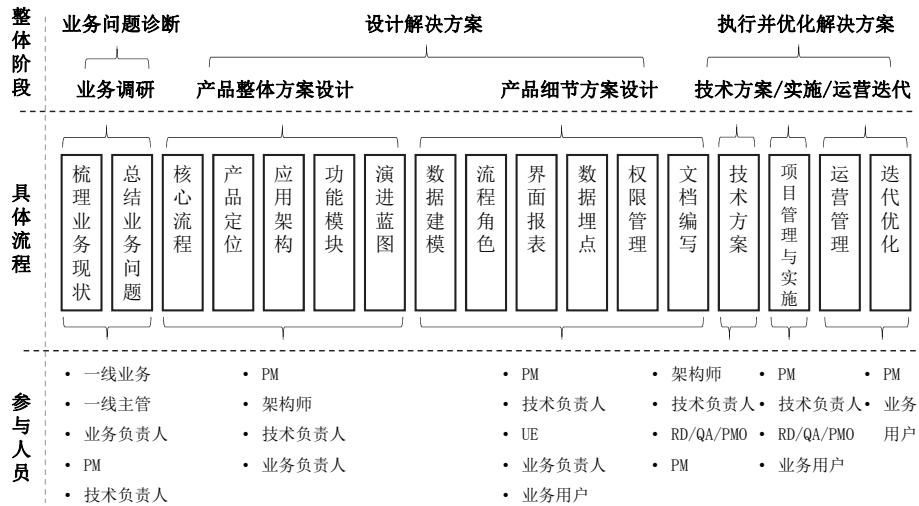


图 3-1 B 端产品的总体建设流程

## 业务调研

业务调研是在产品设计之前要开展的非常重要的准备工作，在这个阶段，产品经理要全面研究并理解业务的现状和规划，挖掘并总结业务问题。

在业务调研阶段，产品经理需要尽可能地用各种手段和工具收集业务关键信息，通过对业务负责人、一线业务人员等角色进行访谈，获取全面的信息；另外，可以邀请技术负责人一起参与业务调研，确保对业务的理解是一致的。

通过业务调研找到关键业务问题，这是设计产品解决方案的核心前提。

我们将在第 4 章详细讲述业务调研。

## 产品整体方案设计

B 端产品整体方案设计讲究体系性、结构性。基于对业务现状与发展方向的理解，产品经理需要和架构师、技术负责人一起，规划产品的功能范围、定位，以及和公司现有产品体系如何融合，形成对后续细节设计有指导意义的整体方案，包含以下方面。

- 核心业务流程：梳理整个业务主干流程，并确定其中哪些环节需要由该产品实现线上化。
- 产品定位：明确该产品有哪些子系统，分别支持哪些业务流程和业务版块。

- 应用架构：考虑该产品和公司现有系统的融合关系。
- 功能模块：基于对业务的理解，抽象出该产品的具体功能模块。
- 演进蓝图：根据业务优先级与发展策略，制订实现各功能模块的计划和节奏。

在产品整体方案设计阶段，业务负责人有必要参与讨论，并且所有参与者需要通过讨论得出一致认可的结果。

我们将在第 5 章详细讲述产品整体方案设计。

## 产品细节方案设计

梳理了核心流程、产品定位、应用架构、功能模块和演进蓝图，相当于完成了万丈高楼建设的规划蓝图；接下来的细节方案设计就要基于蓝图，逐一分析业务细节，设计产品的具体功能。

数据建模，也叫业务建模或领域建模，是细节方案设计中最重要的环节，是保证产品设计严谨可行的关键工作。只有基于对业务的理解，抽象出合理且灵活的数据模型，才能设计出有持续灵活性和扩展性的应用系统。在后续的学习中，你会慢慢理解为什么数据建模是产品细节设计成功的基石。

角色与流程设计会涉及业务团队的组织架构和岗位编制，需要产品经理与业务负责人一起讨论决定。

界面与报表是业务用户直接看到的部分，在设计时最好能提供可以体验的交互界面，让业务用户提前感受并反馈意见，减少不必要的返工。

我们将在第 6 章详细讲述产品细节方案设计。

## 技术方案设计

产品的整体方案、细节方案都设计好后，就需要技术人员做技术方案设计了，从而保证软件系统在正确的技术选型和合理的技术架构下进行编码开发工作。产品经理一般不需要直接参与技术方案设计，但还是有必要理解相关技术知识。

我们将在第 7 章讲述技术方案设计相关的内容，并梳理产品经理应该具备的技术知识体系。

## 项目管理与实施

技术方案设计完毕，接下来就要进入具体的开发实施环节了。

B 端产品往往涉及多个业务部门，需要多个业务系统的跨端配合，如何推进跨端项目？如何保证项目如期高质量交付？做好项目管理是关键：完善的项目管理机制可以保证实施环节顺利进行；相反，如果项目管理混乱，任意变更需求、扩大项目范围，就会导致项目无限延期。

我们将在第 8 章详细讲述互联网产品的管理，以及项目推进执行的相关问题。

## 运营迭代

新系统上线后，产品经理要和业务人员一起参与产品的运营迭代工作，包括宣传、推广、使用效果分析、问题和反馈意见的收集，以及持续的迭代优化。

B 端产品的运营管理有一定独特性，因为 B 端产品都存在对应的业务方，而业务部门都会设立业务运营团队。在 B 端产品领域，产品经理、产品运营、业务运营三者的工作职责往往有所重叠，各自的工作内容该怎样分配？协作关系该怎样处理？处理好这些问题会让你的工作事半功倍。

我们将在第 9 章详细讲述 B 端产品的运营管理。

对于迭代优化工作，B 端产品也有自己的特点，我们需要管理好需求、分配好研发资源、选择合适的迭代模式。

我们将在第 10 章详细讲述 B 端产品的迭代优化。

## 3.2 B 端产品与 C 端产品建设流程的区别

在产品从 0 到 1 的建设过程中，B 端产品和 C 端产品在很多环节上是完全不同的。图 3-2 将两者的建设流程对比呈现出来了，我们可以清晰地看出两者的异同点。其中，C 端产品的建设流程是根据经验总结抽象出的常见流程，不同的需求和背景下的流程可能略有不同。

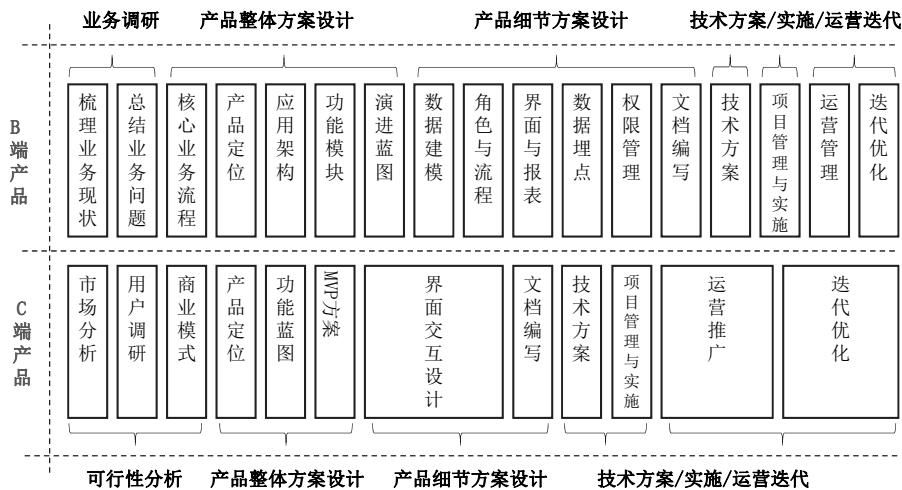


图 3-2 B 端产品和 C 端产品建设流程对比

从图 3-2 可以看出，B 端和 C 端产品的建设流程很大不同，具体体现在如下方面。

## 设计起点不同

进行产品设计之前都需要进行调研，这是设计的起点。因为 B 端和 C 端产品的定位、目标完全不同，所以两者的设计起点不同：

- B 端产品是为了解决业务问题而设计的，设计的起点是进行业务调研，研究业务问题。
- C 端产品要实现公司商业模式的落地，承载着公司的商业目标，设计的起点是对商业模式本身的分析与研究，包括市场分析、客户群分析等。

当然，如果是一家 SaaS 软件公司，设计的 B 端产品要卖给具体客户，那么设计的起点就和 C 端产品一样，是进行商业分析，而不是进行业务调研。

## MVP 思路不同

MVP（最小可行产品，Minimum Viable Product）是《精益创业》一书中提出的产品理念，在互联网公司中被广泛接受并实践，简单讲就是用最小的投入去验证业务，通过快速迭代逐步优化。

在建设 B 端和 C 端产品时，大的原则是类似的，都是先做加法，即充分讨论、穷举所有需求和可能性；然后再做减法，选出最核心的需求点；最后设计具体方案并将

其落地，用最短的时间和最低的成本支持业务启动。

但是在选取最小功能集合（或最小可行产品）时，B 端和 C 端产品的区别很大：

- B 端产品要支持业务整体运作，所以在选取最小功能集合时，即便再简化，也要保证一个核心业务流程的运转，因此 B 端 MVP 往往是一个具备一定复杂度的系统，不可能是一个或几个功能点。
- C 端产品需要解决用户的痛点，需要挑选一个核心痛点去打动用户，如果核心痛点定位错误，就会导致验证失败。所以在选取最小功能集合时，C 端产品要聚焦用户的核心痛点，C 端 MVP 可能只包含一两个功能点。

## 细节设计不同

两类产品在细节设计上的关注点可谓完全不同。

- B 端产品面临复杂的业务场景和用户场景，因此进行细节设计时，必须关注建模、抽象、角色、权限等问题。
- C 端产品面临的场景相对单一，并且使用者是相对独立的单个用户，因此不用关心角色、权限管理，而要关注用户的体验，需要在交互设计上投入很大精力。

## 对运营的依赖程度不同

相对 B 端产品来说，C 端产品对运营更依赖：

- B 端产品上线后，要进行全员宣导培训，产品运营工作相对简单。可以说，B 端产品上线后肯定有用户使用，因为公司会要求自己的业务团队使用新产品。
- C 端产品上线只是走完了万里长征的第一步，接下来需要运营团队进行持续推广，并且通过快速迭代迅速优化产品，响应用户需求。可以说 C 端产品上线后还要靠运营团队继续奋战，才可能走向成功。在图 3-2 中，我们将 C 端产品运营迭代的过程绘制得更长一些，以体现运营工作对 C 端产品的重要性。

## 3.3 案例：M 电商公司的渠道分销产品设计

通过 3.1 节的讲解，大家对 B 端产品的整体建设流程应该有了初步认识。接下来，我们会通过一个贯穿设计篇和管理篇的大案例，按照图 3-1 中的流程，带领大家一步

步搭建一个 B 端产品。首先，我们来介绍一下案例的背景。

### 3.3.1 案例背景与目标

#### 背景

M 集团是一家经营了十几年的成功企业，旗下拥有零售连锁超市、生鲜电商、金融理财多条业务线，业务发展良好，系统建设成熟。

M 公司是 M 集团下属的电商公司，成立 5 年，主营生鲜商品，以 C 端客户为主，业务稳定。

M 公司在三个月前尝试开展分销业务<sup>1</sup>，成立销售团队，开发分销商合作伙伴。分销业务的目标客户是大型的餐饮连锁集团，以及大型生鲜分销商等企业级客户。

业务试点在北京、上海开展，三个月以来业务发展迅速。目前分销业务月流水 50 万元，以每月 20% 的增幅快速发展。但是，在高速发展中，若干流程、管理、风险问题越来越突出。

#### 诉求

由于分销业务发展迅速，现急需配套的软件系统来提升业务效率，控制经营风险。

#### 评估

经管理层评估，公司决定投入研发资源建设软件系统，支撑分销业务发展。项目期间 CTO 全力提供资源支持。

#### 目标

在 2~3 个月的时间内搭建一套分销业务平台，至少支撑分销业务在未来 2 年内的高速发展，有效地提升效率、控制经营风险。

以上就是案例的大概背景。之所以选择一家集团企业下属电商公司的分销业务作为案例，是因为它非常有代表性：

---

<sup>1</sup> 分销渠道是非常重要的商品销售渠道之一，它介于商品的生产者和终端消费者之间，帮助生产者将商品批量推广到下游经销商或终端客户。分销业务具有薄利多销的特点。

- 首先，M 集团是一家成熟集团，拥有完善的应用架构，大家可以了解如何将新设计的产品与公司现有产品架构融合。
- 其次，分销业务场景具备足够的复杂性，既要支持公司对客户的运营管理，又要支持客户的自主管理，涉及的系统具备比较全面的功能。
- 最后，分销业务模式涉及复杂的多层级子母账号<sup>1</sup>管理和组织机构管理，这是 B 端产品设计中的典型问题，也是设计的难点。

为了让大家更加聚焦产品设计而非业务规则，我们将会对一些无关紧要的业务细节做处理，大家阅读时请重点关注业务分析、产品设计的思路，忽略案例中可能存在的数据和流程上的小瑕疵。

### 3.3.2 制订工作计划

假设你是产品经理，接到公司安排的这项任务：在 2~3 个月内搭建一套支撑分销业务的软件系统，你会怎样入手呢？这是一项大任务，产品经理首先需要做的是梳理工作思路，拆解任务，并制订时间计划，只有严格遵循时间计划执行工作，才能保证整体工作有序展开，如期落地。

在制订项目计划时需要略微卡紧节奏，我们按照两个月来安排，这样能够为各种意外情况留一些应对时间。假设目标是两个月时系统一期上线，进行倒逼排期：最后一到两周联调，往前三到四周开发，最开始的三周完成业务调研、方案设计工作（如图 3-3 所示）。当然，这只是产品经理初步安排的时间表，接下来需要尽快了解更多信息和情况，才能做出更合理细致的时间预估。

---

<sup>1</sup> 母账号（也叫根账号）是供客户或机构用户使用的最高级别的管理员账号；子账号一般是分配给具体业务人员使用的个人账号。

序号	任务	负责人	进度	前置任务	Jun				July			
					W 1	W 2	W 3	W 4	W 1	W 2	W 3	W 4
1	业务调研	产品经理, 业务人员	100%	-								
2	系统整体方案设计	产品经理, 架构师	10%	1								
3	系统细节方案设计	产品经理	0%	2								
4	技术方案设计	研发人员	0%	3								
5	xxx模块开发	研发人员	0%	4								
6	yyy模块开发	研发人员	0%	4								
7	联调	研发人员, 测试人员	0%	5、6								
8	上线	研发人员, 运维人员	0%	7								

图 3-3 分销平台项目整体计划表

图 3-3 是一个简化版的甘特图，虽然简单，但非常实用。左侧列出了拆分后的具体任务、负责人、进度，右侧列出了项目计划的时间周期，采用了比较粗的周粒度来计划时间，这样，任务的先后顺序、时间计划就都清晰了。在刚开始的阶段，这个表格可能只是产品经理自己的行动计划，并没有向团队或项目组展示，但是它可以让产品经理对事情有基本的判断和预期。随着工作的深入，工作表会被拆分得越来越细，甚至细化到天，每日跟踪计划和风险点，保证项目如期进行。

当我们接手一件比较复杂的工作时，制订明确的工作计划是一种良好的工作习惯。即便是个人管理使用，梳理思路并拆解出关键任务和计划的时间点，也是非常有必要的，会让自己感到踏实，有节奏感。如果没有进行时间管理，则很容易感到焦头烂额，难以控制。因此，不论做什么事情，都应该先从总体上理清思路，列好时间计划。

制订好工作计划，我们进入业务调研环节。

## 第 05 章

# B 端产品的整体方案设计

通过业务调研，我们已经对业务脉络有了较好的了解和掌握，对业务问题有了比较准确的判断和总结。现在，需要从整体上构思产品的解决方案了。

B 端产品的整体方案设计需要遵循自顶向下的设计思路，可以依次设计核心业务流程、产品定位、应用架构、功能模块、演进蓝图，从抽象到具体，逐步勾勒出 B 端产品的轮廓。这些是后续细节设计的指导性方针，是细化设计的基础。

## 5.1 核心业务流程

从核心业务流程切入产品设计，是开展整个设计工作的非常好的起点。核心业务流程代表业务的主干脉络，需要思考业务的各个参与方、涉及的系统。

根据第 4 章对 M 公司的业务调研，我们总结出当前紧迫的业务诉求包括客户自主下单、自动定价，以及支持客户对多门店分别定价与下单。因此我们考虑调整业务流程，由运营人员将客户信息录入系统，客户具备一个初始账号后，可以自主管理下属的多个门店和子账号。

因为 M 公司可能和客户总部签署了框架协议，具体合作关系又由客户的下属分公司或门店分别谈判，所以价格系数表要针对每个门店分别设置，以保证定价的灵活性。因此当客户创建新门店后，还需要由运营人员设置一次价格系数表，之后，门店采购人员就可以通过关联的子账号进行下单了。

下单之后的环节保持之前的运作流程，不做改变。

综上，我们可以绘制出设计的跨部门核心业务流程图，如图 5-1 所示。

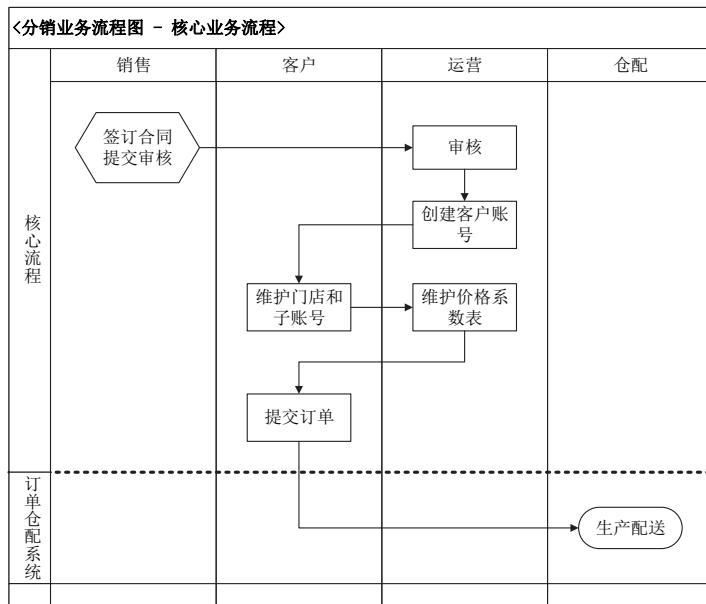


图 5-1 M 公司分销业务的跨部门核心业务流程图

经过和分销业务负责人沟通，因为目前销售人员数量有限，暂不需要设计给分销业务销售人员使用的 CRM 系统，客户线索管理、客户资质审批与合同签订审核，都可以继续保持线下运作。

因此，我们考虑将客户签约作为一个分界点，签约之前的环节依然线下运作，而为签约之后的环节设计一套全新的系统，支持后续业务流转。我们将这套系统暂且称为分销系统，将图 5-1 中的流程图略修改，体现出包含新系统的新业务流程图，如图 5-2 所示。从图中可以清晰地看出一套全新的业务系统对分销业务的支持，及其在主营业务流程中的位置。

通过对核心流程的梳理，以及明确其中哪些环节需要线上化支持，分销系统的轮廓初现。

此时，我们暂时不需要深入细节，而要基于这个初现的轮廓，进一步思考分销产品的顶层设计。

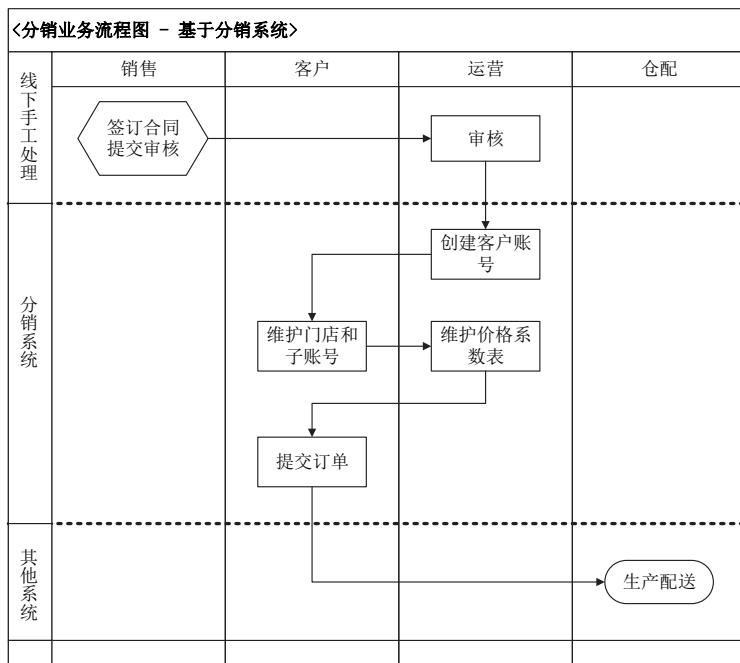


图 5-2 M 公司基于分销系统的核心业务流程图

## 5.2 产品定位

产品定位是对产品概要性的总结和陈述，简明扼要地描述产品对业务的支持范围，或总体的功能目标。产品定位要说清楚产品针对谁提供什么支持。

通过对 M 公司分销业务的核心流程梳理，我们已经对分销系统在整个系统中的位置有了初步判断，现在，我们进一步思考更精确的形态和定位。

首先在分销业务中，客户需要一个快速下单的工具，可以提供一个手机版商城 C 端。考虑到投入产出比，公司决定通过 H5 来实现，而不是 App。H5 所写的网站具有独立域名，外网可访问。

其次，需要为客户提供一套方便操作的管理后台，因为涉及大量的商品定价编辑、处理，账号、门店管理等功能，所以考虑 PC 版本实现，暂不支持手机版。

最后，考虑到客户管理员和 M 公司管理人员的管理诉求不同，操作功能和页面差异较大，所以决定将管理后台拆解为两个独立的系统：给客户管理员使用的客户管理后台具备独立域名，外网可访问；给 M 公司管理人员（和运营人员）使用的运营管理

后台也具备独立域名，但仅限内网访问。

经过以上分析，我们进一步将分销系统拆分为三个独立系统，每个系统的定位不同：

- 分销商城前台（移动端，通过 H5 实现）：为分销客户提供下单功能。
- 客户管理后台（PC 端）：为分销客户提供子账号管理、门店管理及业务辅助功能。
- 运营管理后台（PC 端）：为分销业务部门提供客户及商品定价管理的业务支持功能。

设计业务系统时的常见问题是，为了省事，或者由于业务部门之间边界模糊、权责界定不清晰，导致很多本应该独立的功能被糅合到一个系统中，这样会造成将来管理的混乱，尤其是系统维护的混乱。理想情况下，独立的业务部门应该由独立系统来支持工作。

举例来说，假设 M 公司分销业务的运营人员和公司订单中心的运营专员是同一拨人，你会考虑在订单系统上扩展功能，来支持分销业务吗？

## 5.3 应用架构设计

所谓应用架构，是指公司所有产品和系统的整体结构和布局，我们在“进阶篇”会详细讲述。任何公司的应用架构都不是随意设计的，有复杂的设计思想蕴含在其中。

在设计一套新系统时，必须考虑如何和公司现有系统架构融合，不同系统的模块之间如何衔接。这项工作复杂度较高，不仅需要有丰富的电商架构经验，而且需要深刻理解业务特点和可能的演进方向，还要熟悉公司目前的系统架构，这样才能快速提炼出相关问题。一般由产品负责人和公司的架构师甚至 CTO 共同讨论确定。

M 集团已经发展了多年，集团的软件体系结构非常成熟，这就意味着在设计一套新的系统或产品时，完全可以复用现有的部分系统或模块，从而快速实现新系统，提高系统建设效率，减少重复开发，更重要的是，保证合理的整体系统架构。

对于新设计的分销平台，该如何和公司现有系统融合呢？

首先，M 公司作为一家成熟的公司，之前已经有面向零售业务的 C 端商城，用于支持个人客户线上购物。那么针对分销业务的客户，我们是单独做一套 C 端商城，还是改造、复用现有的 C 端商城呢？经过分析评估，个人客户和分销客户（属于企业客

户)需要的功能差别较大,因而两套商城的整体区别较大,如果通过对原有商城进行改造来支持分销业务,需投入的工时会很多,甚至可能比新开发一套系统还要多,而且还会影响主营业务系统的健壮性,因此公司最终决定做一套新的C端商城(分销商城前台)来支持分销业务。

其次,我们要思考的问题是如何维护、管理客户数据,是在分销平台中独立管理,还是通过公司现有的客户资料管理模块管理?对于M公司来讲,现有C端客户资料全部存储在客户主数据MDM(Master Data Management)系统中,我们认为无论是C端客户信息,还是分销业务的B端客户信息,都是企业的核心资产,应该采用集团统一管理的方式,因此决定通过改造现有客户主数据系统,支持分销业务的客户信息存储和管理。

接着我们要思考如何管理客户账号体系。账号是用户登录系统的凭证,对于企业来讲,账号和客户是两个概念,一个客户可能有多个账号,但一个账号只会对应一个客户。比较成熟的企业都具备一套成熟的用户管理中心(Passport)系统,实现统一账号管理。经过思考,我们打算将现有Passport系统升级,从单一账号体系升级为子母账号体系,支持分销渠道的企业客户账号管理。

然后我们考虑如何连接订单和仓配系统。公司已经有一套成熟的订单中心,基于订单中心可以完成正逆向交易操作及财务处理,且目前已经在支持分销业务手工下单模式的作业处理。因为分销业务售卖的商品和C端业务售卖的商品来源是相同的,所以订单中心能够完美支持分销业务,因此决定分销平台不单独开发订单中心,而直接把交易发送给现有订单中心,由它作为桥梁,连接后续生产配送环节。这样就实现了分销平台和仓配系统的完美解耦。

关于分销平台的商品管理,完全可以复用C端的商品中心,只需要对价格模块做定制开发。

最后,我们考虑后端系统的权限管理,以及分销商城前台的支付问题。由于M公司现有权限管理系统(Auth)与支付平台(Pay)都是基于服务化建设思路实现的完善解决方案,因此可以作为基础服务快速支持新系统,无须重复开发。

至此,我们梳理并确定了分销平台和公司现有架构的融合关系,确定了系统的复用方案。我们将分销平台分为三个子系统,将其绘制到公司整体应用架构图中,如图5-3所示。深灰色矩形是新增的系统,浅灰色矩形是原平台中需要结合本次项目进行升级修改的系统,白色矩形是原平台中不受影响的系统。关于这张架构图代表的含义、形成的原因、分层的目的,在“进阶篇”会详细讲解。

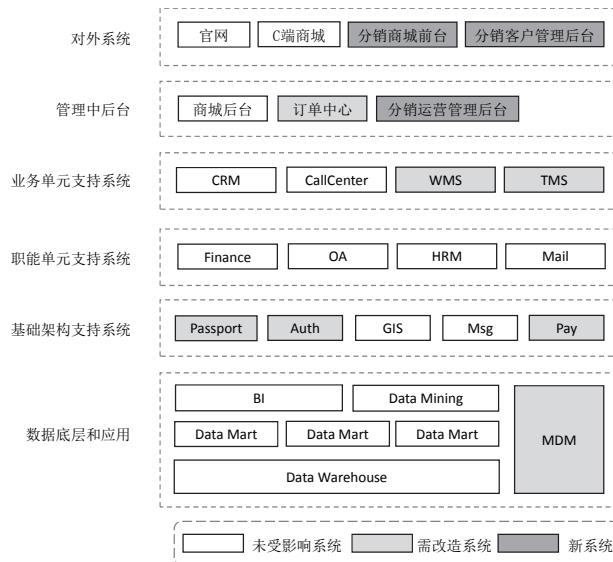


图 5-3 支持分销系统的公司整体应用架构图

## 5.4 功能模块设计

明确了应用架构，以及需要新建或改造的系统之后，我们需要进一步细化，为每个系统设计功能模块。这个系统应用于哪些业务场景？用户可能在系统中做的操作有哪些？通过思考这些问题来抽象出需要具备的功能模块。产品经理设计的功能模块代表了其对业务本质诉求的理解和提炼，蕴含了他对业务、系统未来发展的期望。

我们常说，系统建设要有规划、有节奏，实际上功能模块图就是一幅完整的规划蓝图，能体现出系统的一二级导航菜单结构，是系统的骨架。结合业务需求实现的每一个具体功能，都是在对骨架不断地填充血肉，让它更真实、更立体、更丰富。

设计功能模块的常见问题是模块层次混乱，以及后来新增功能的随意摆放，这都会造成用户使用系统时产生困惑，同时也会导致开发人员编码设计的混乱。下面我们将为分销业务设计功能模块。

通过自顶向下的分析思路，我们明确了分销业务的三个独立系统（分销商城前台、分销客户管理后台、分销运营管理后台，见图 5-3），以及三个系统与公司整体架构的融合关系。接下来，我们进一步拆解设计，每个独立系统应该具备哪些功能或模块？我们把能想到的功能集合，现在或未来可能需要的功能列表都列出来，这是一个产品做加法的过程。

## 分销商城前台

分销商城前台即客户下单的 H5 工具，是一个经典的电商 C 端系统，分销客户需要在上面完成下单购买操作，也需要完成自我管理（例如对下属门店的管理，对发票、售后的管理），因此主要包括购买流程和个人中心两大部分。从购买流程的角度考虑，商城需要具备以下功能或模块：首页、搜索、推荐、列表页、详情页、购物车、结算页、收银台。个人中心要包括订单管理模块、账号管理模块、售后管理模块等；针对分销业务的特殊诉求，还需要包括门店维护、对账管理等模块。分销商城前台的功能模块如图 5-4 所示。



图 5-4 分销商城前台的功能模块图

## 分销客户管理后台

分销客户管理后台是给分销客户管理员使用的管理后台，主要用来管理下属门店和子账号；还需要随时了解下属门店和子账号的经营情况，因而需要查询所有下属门店和子账号的数据；此外还需要进行统一的财务管理。因此分销客户管理后台一共包括下面三个大模块，其功能模块如图 5-5 所示。

- 客户管理模块，支持子账号管理与门店管理。
- 综合查询模块，实现所有可能的查询与信息检索诉求，包括门店报表、订单查询、综合报表、售后查询。
- 财务管理模块，支持基本的发票管理、对账管理，以及分销业务特有的预付款管理。

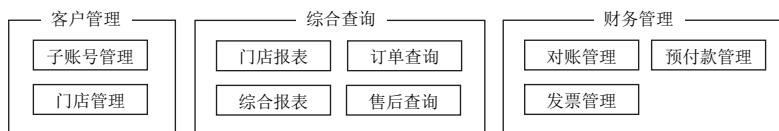


图 5-5 分销客户管理后台的功能模块图

## 分销运营管理后台

分销运营管理后台是支持 M 公司分销业务的核心业务系统，同时也是一套典型的电商管理后台。典型的电商管理后台需要具备商品定价管理、财务管理、风控管理、运营管理、客户管理、报表管理几大模块，另外，针对案例中的分销业务，还需要具备账期管理模块，其功能模块如图 5-6 所示。

- 商品定价管理模块：一般支持商品管理、价格管理。根据 5.3 节的分析，在 M 公司的分销业务中，其商品管理模块将完全复用 C 端业务的商品中心；其价格管理将通过价格系数设置模块和门店报价管理模块完成；商品的基本定价数据将从 C 端业务的价格中心获取，然后在分销平台维护价格系数表和门店报价单，从而计算针对不同客户和门店的售价。
- 财务管理和账期管理：基于前期业务调研，我们明确分销业务要支持账期和预付款管理，所以相对应的账期回款监控、预付款管理都是必备功能。
- 报表管理：报表管理模块将提供各类分析报表，实现对业务运作情况的监控和诊断。
- 风控管理、运营管理：在这两个一级模块中还可以实现定价风控、订单风控、CMS（内容管理）以及消息中心等，这里不再一一介绍。

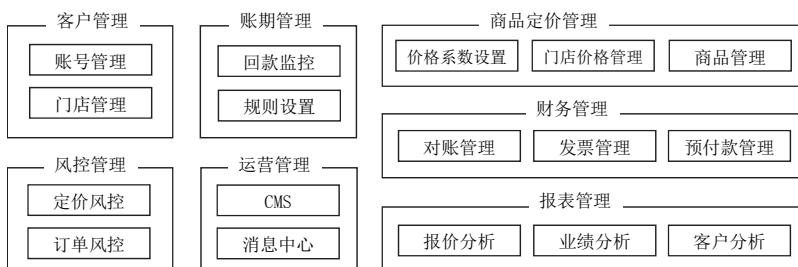


图 5-6 分销运营管理后台的功能模块图

## 5.5 演进蓝图设计

通过绘制系统的功能模块图，可以明确业务和系统的规划脉络。将能想到的功能集合都列出来，这是一个做加法的过程。但是我们不可能一次实现全部功能，而要根据业务优先级，拆分成几期来完成，所以接下来需要做减法：确认产品的功能规划与

实现节奏，就是常说的演进蓝图（Roadmap）。

在对 M 公司的业务调研中，我们不仅列出了需求，而且明确了需求的优先级（参见 4.5.2 节）。根据优先级，以及前面绘制的分销平台三个系统的完整功能模块（图 5-4、图 5-5、图 5-6），我们计划将分销平台分为三期实现，其演进蓝图如图 5-7 所示。



图 5-7 分销平台功能规划演进蓝图

一期项目聚焦解决最基本的业务流程线上化问题及核心痛点（例如对账功能），在图 5-7 中用白色矩形表示。一期项目要实现哪些功能？有一个原则可以参考：凡是可以通过手工处理和解决的问题，都暂时不做系统支持。例如，报表管理功能可以通过定期运行 SQL 语句实现；价格系数设置功能的使用频率低，可以由 RD 在后台改数据库完成；缺少搜索、推荐功能并不会对客户下单的效率产生明显影响，因为根据调研，目前每个客户维护的 SKU 数量最多也不过 20 个。

二期项目聚焦于解决部分特殊业务刚需的诉求。对于 M 公司的分销业务，需要支持预付款模式、账期模式、发票管理，如果时间允许，还可以实现报表查询的若干功能。在图 5-7 中用浅灰色矩形表示。

三期项目聚焦风险控制，并强化运营功能，在图 5-7 中用深灰色矩形表示。一般来讲，很多互联网公司初期都会聚焦于业务本身，提升 GMV（成交总额）、验证可行性，不会太在意成本和风险控制。当业务达到一定规模时，则必须引入系统风控机制，实现事前、事中、事后的风险控制。

此外，基于本案例 B2B 业务的特点，我们在设计中并没有考虑太多的 C 端功能。实际上 C 端功能只需要保证分销客户能够轻松下单，并做一些简单的运营、通知即可。

随着设计的深入，以及业务的开展、变化，功能模块可能需要修正和调整，但只要业务的本质模式没有变化，功能模块就不应该出现结构性的改动。功能模块图和演进蓝图代表的是概要性方案，指明了整体的产品方向，是后续细化设计的指引和准则。

设计软件产品时必须遵循自顶向下的设计思路，这是非常重要的，相信大家已经有了初步的感觉。此外，在互联网产品圈中很流行的用户体验五要素及其提出的五个设计层次（表现层、框架层、结构层、范围层、战略层），也是一种自顶向下、由粗到细的设计思路，感兴趣的读者可自行查阅。