ToCode 代码生成平台
需求分析及产品设计方案

（第一版）

郑琳

2023 年 8 月 18 日

目录

一、项目背景

1.1 关于软件项目

一个软件项目实施从制订实施计划开始，到完整上线，是较为漫长的过程，同一类产品，针对不同的企业、不同的产品定位、不同的时间段，可能都会有截然不同的客制化需求。

针对不同的客制化需求，就需要进行软件定制开发，软件开发作为高度依赖开发人员的虚拟行为，存在着成本难以控制和质量难以保证两大难题。

1.2 分层架构与模块化开发

为了降低开发成本，让产品拥有更好的扩展性、可维护性，分层架构与模块化开发进入到广大开发人员的视野当中。

将一个项目，根据不同的外在表现或是工作重点，划分不同的层；同样，一个项目可以划分若干不同的模块，将耦合度高的内容划分到同一模块中。

分层架构与模块化可以很好的将一个项目以非常小的颗粒度进行拆解管理，不同的开发人员只负责其中的一部分内容，因为大多数时候只需要关注自己负责的内容，开发人员的工作效率可以大幅度提高，质量也可以得到保证。

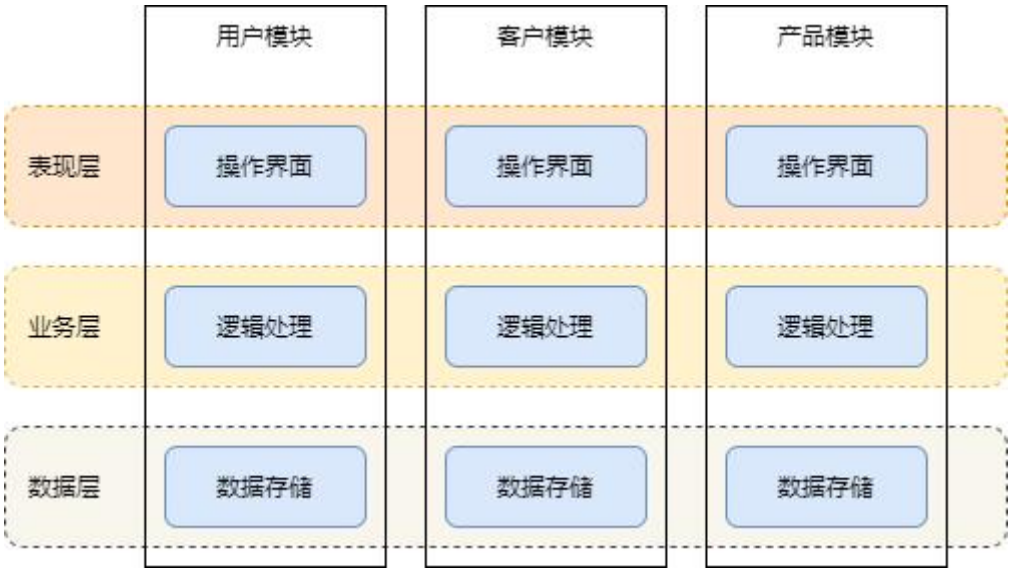

图 1-1 产品分层与模块化

1.3 标准化产品+客制化

随着项目经验的不断增加，很多人发现产品的大部分功能都具有共性，于是

逐渐演化出标准产品+客制化这种形式，来大幅度降低重复功能的代码编写，从而缩短开发周期。

同时，因为开发的内容变少，开发人员所专注的工作内容也响应减少，开发人员的工作量大幅度减少，质量也可以得到保证。

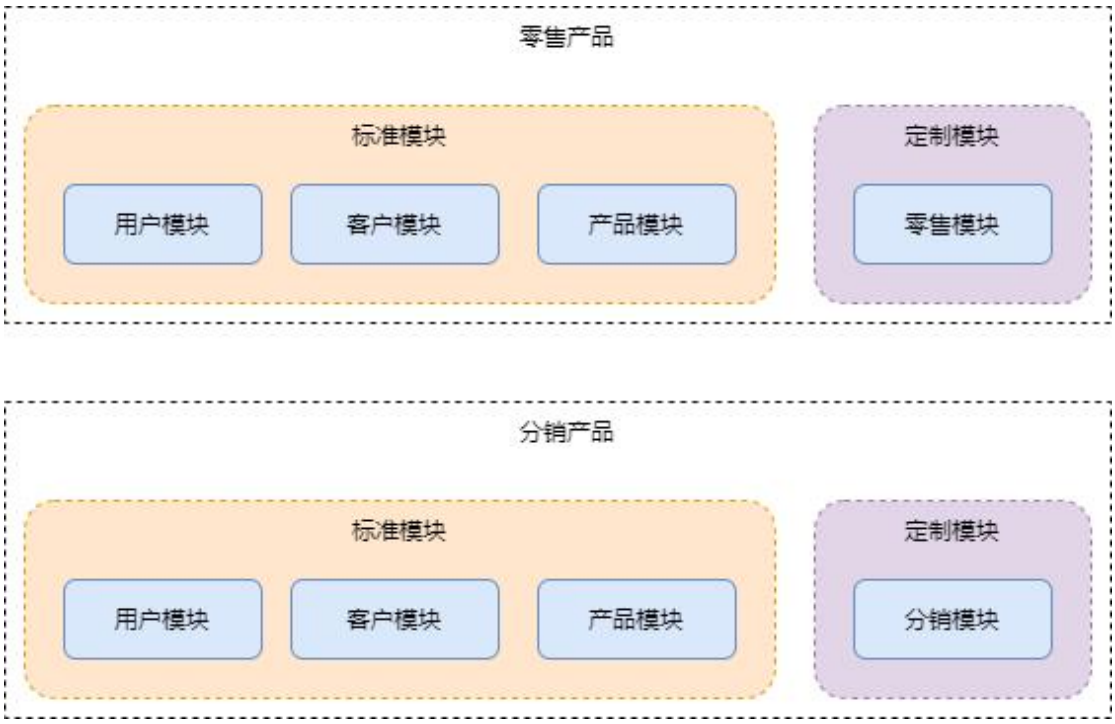

图 1-2 标准化产品+客制化示例

1.4 低代码平台

随着大量客制化软件的实施后，越来越多的人发现很多客制化功能都是在一定的可遇见范围，于是，低代码平台进入大家的视野。

低代码开发平台是无需编码（0 代码）或通过少量代码就可以快速生成应用程序的开发平台。

在低代码开发平台中，大多数功能可以通过鼠标拖拽或者表单填写的方式实现，这种平台最大的优势，就是大幅度减少了产品对开发人员的依赖，转而让通过培训后的非开发人员，也可以参与产品功能的生成当中。

二、代码生成平台

2.1 低代码的困境

虽然低代码平台看上去非常美好，但真正的低代码平台需要考虑的东西非常多，目前没有一家产品可以做到完美的程度，能提供的功能大多非常有限，并且因为高度封装与较为闭环的开发生态，导致一般的开发人员无法参与开发，遇到复杂的需求可能直接面临束手无策的境地。

2.2 标准化产品+代码生成

无论从哪个角度看，低代码平台都存在着很多需要平衡舍取的现状。

那如果有一种基于程序员的代码生成平台，采用与低代码平台类似的操作方式，鼠标拖拽或表单操作，可以生成标准产品所需的扩展插件或者是可供修改的源代码。

代码生成平台的优势将会非常明显：普通人员可以直接使用平台生成简单的功能插件给标准产品使用；复杂的功能可以通过先生成源代码，再交予开发人员进行更深度的开发。

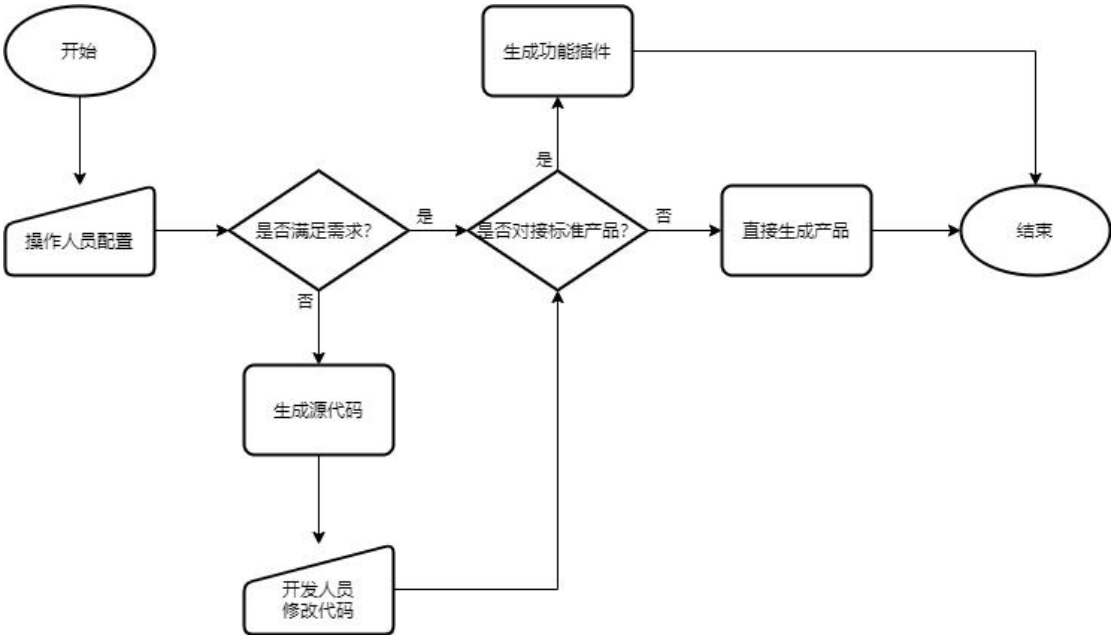

图 2-1 代码生成流程

三、总体方案

ToCode 是一款基于配置项的通用代码生成平台。

ToCode 通过对接数据库、配置项等方式，一键生成多种编程语言的源代码，也支持部分语言的一键编译出动态库。

3.1 项目化管理

多项目管理，一键切换

针对不同项目，不同的需求，产生不同的程序代码，是基本要求。

为了更加清晰的管理项目，项目还需要支持分类管理。

一个项目，还可包含若干的架构分层。

3.2 模板化管理

使用代码模板定义代码段

不同的项目，同一个的功能可能需要生成不同的代码，为了达到更好的兼容性，使用代码模板可以很好的解决这个问题。

不同的项目将会独立管理代码模板。

一个代码模板可以包含多个代码段。

3.3 基于功能的代码生成

功能是最小的管理维度

按照功能划分来进行数据表、业务逻辑的管理。

一个项目，可以包含不同的模块；一个模块，可以包含不同的功能。

同一功能，可根据不同的架构分层进行代码生成。

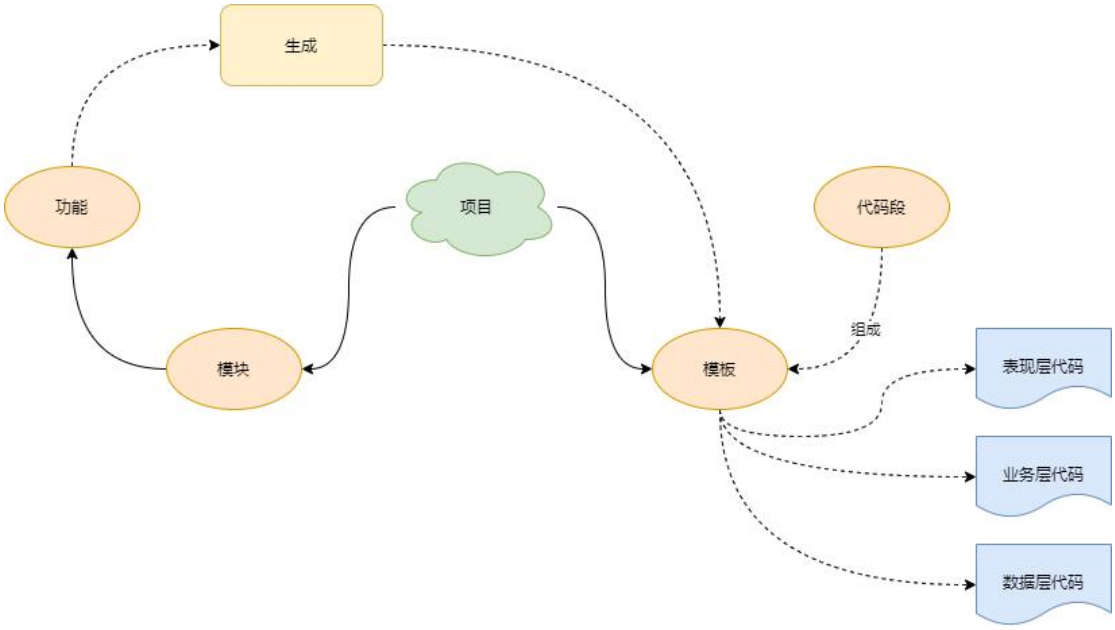

图 3-1 实体关系图

3.4 支持外部对接

可扩展性才能让软件拥有更多的应用场景

使用 RESTful API 提供对外接口。

四、详细设计

4.1 总体业务流程

明确了总体方案后，大致的业务流程呼之欲出：

- 维护项目分类
- 在项目分类中维护项目
- 在项目中维护模板信息
- 在模板中维护代码段信息
- 在项目中维护模块信息
- 在模块中维护功能信息
- 在功能中添加实例信息、关联模板
- 根据功能生成代码

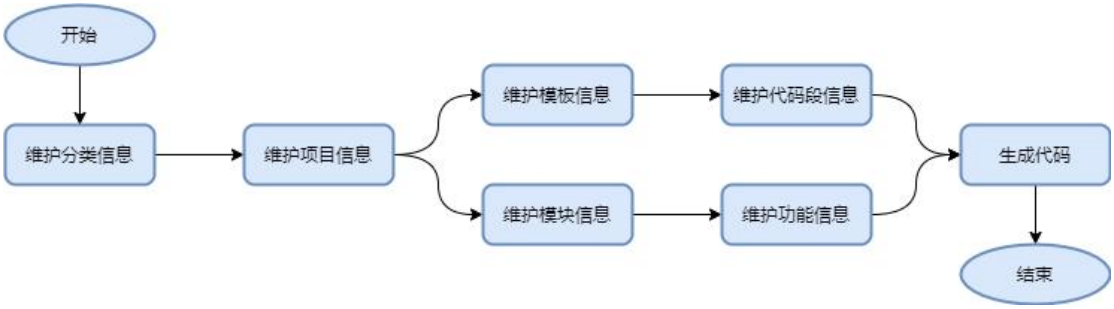

图 4-1 总体业务流程

4.2 总体模块设计

从业务流程中，基本可以将整个系统划分为项目模块、模板模块、功能模块和生成模块四大模块。

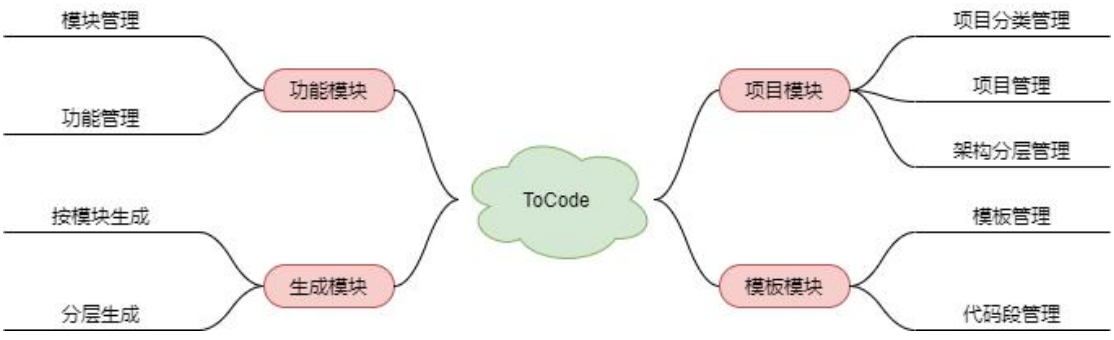

图 4-2 总体模块设计

4.3 详细模块说明

4.3.1 项目模块

项目模块主要分为项目分类管理、项目管理、架构分层管理。

一个项目分类对应多个项目。

一个项目对应多个架构分层。

4.3.2 模板模块

模板模块主要分为模板管理和代码段管理。

模板从属于项目，一个项目可配置多个模板。

模板需要关联架构分层，一个模板对应一个架构分层。

一个模板对应多个代码段。

4.3.3 功能模块

功能模块主要分为模块管理和功能管理。

模块从属于项目，一个项目可关联多个模块。

一个模块可对应多个功能。

一个功能下可关联多套模板。

一个功能下可维护多个实例信息。

4.3.4 生成模块

生成模块主要负责代码生成与构建。

生成的入口在功能上，每个功能都具有独立的生成操作。

生成时加载功能下的所有模板信息，并根据模板关联的架构分层生成对应的代码。

每个实例支持 Sql 脚本生成。

支持构建的项目，可按项目构建 dll 文件。

五、方案实施

整个方案分为三块内容，可分开实施：

5.1 后端开发

服务端主体

后端开发需要实施的模块如下：

模块	说明
项目模块	项目分类信息管理 项目信息管理
模板模块	模板管理 代码段管理
功能模块	模块管理 功能管理 功能实例管理 功能模板管理
生成模块	代码文件生成 代码文件预览 Sql 脚本生成 Sql 脚本预览 项目构建 项目构建 dll 下载

5.2 前端开发

操作界面

前端开发需要实施的模块如下：

模块	说明
主界面	界面布局 功能列表
项目模块	项目分类管理界面 项目管理界面
模板模块	模板管理界面

	代码段管理界面
功能模块	模块管理界面 功能管理界面 功能实例管理界面 功能模板管理界面
生成模块	生成接口对接 代码预览 脚本生成接口对接与预览 构建接口对接

5.3 扩展开发

客户端

扩展开发需要实施的模块如下：

模块	说明
项目模块	项目信息获取 项目目录管理
生成模块	代码下载 代码拷贝 项目构建 dll 下载