

iceworks 4.0：我们是如何做用户调研和产品规划的

01. 产品背景

02. 用户调研

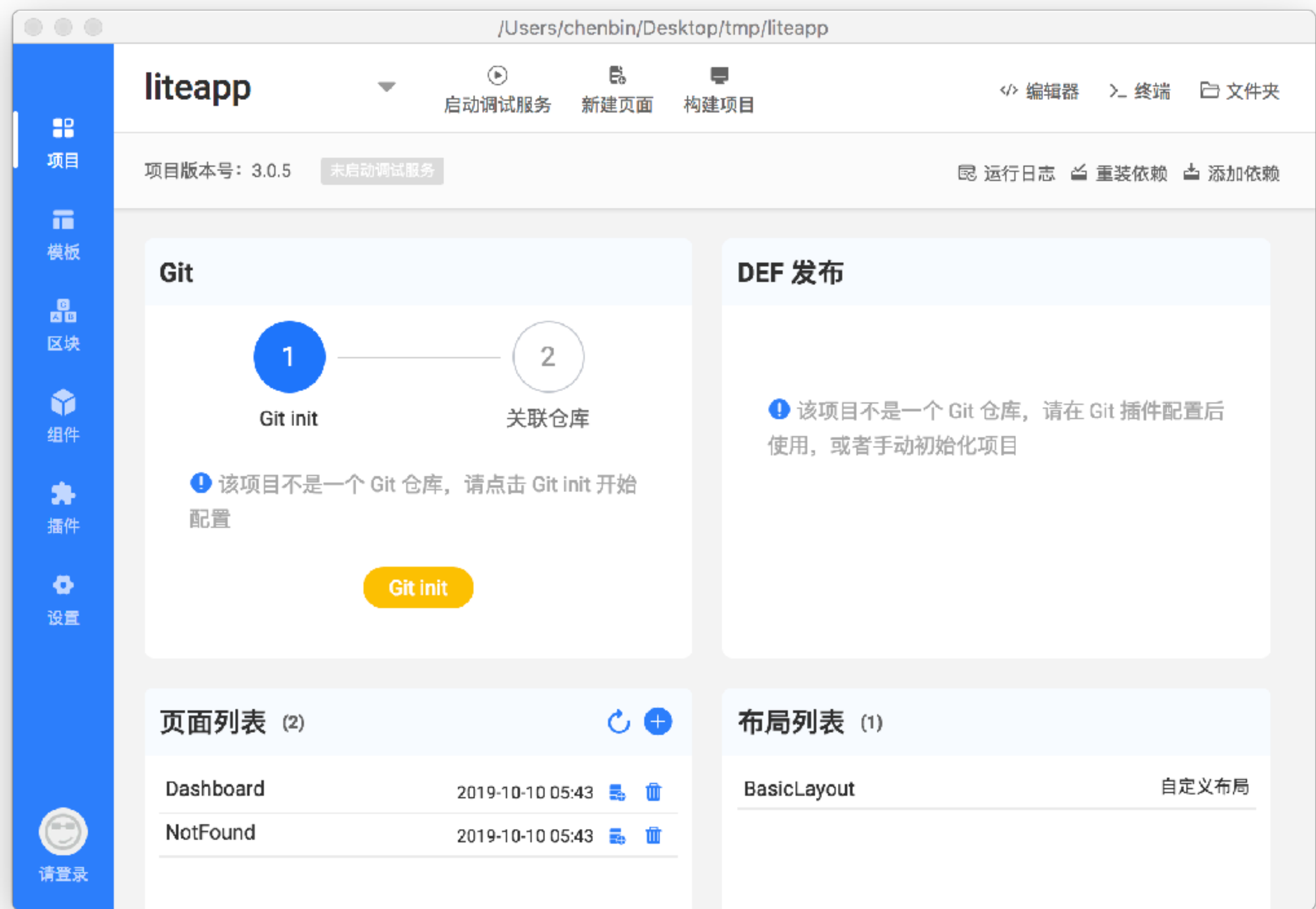
03. 产品规划

01. 产品背景

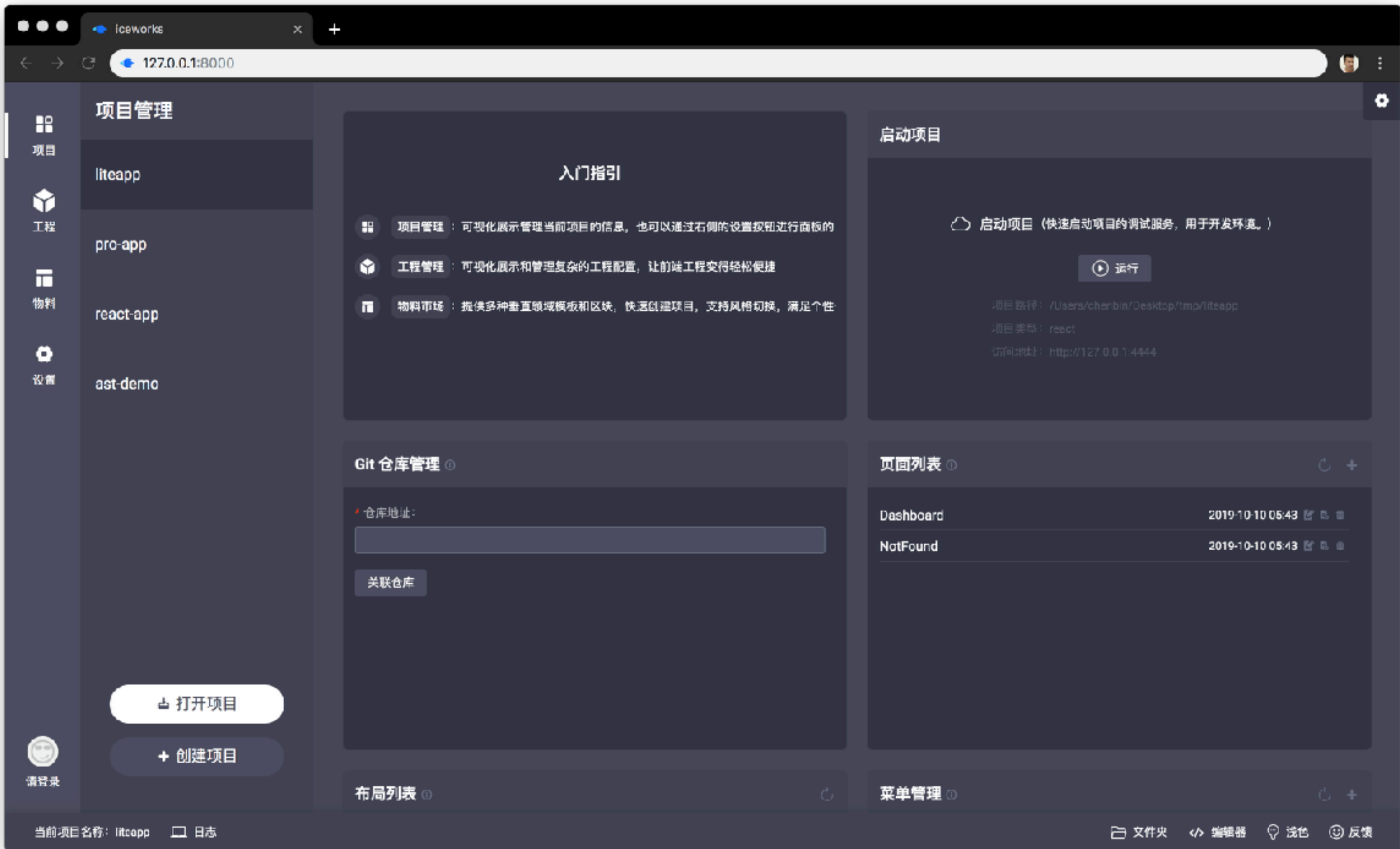
产品背景 - 发展历程



产品背景 - 产品现状



iceworks 2.0

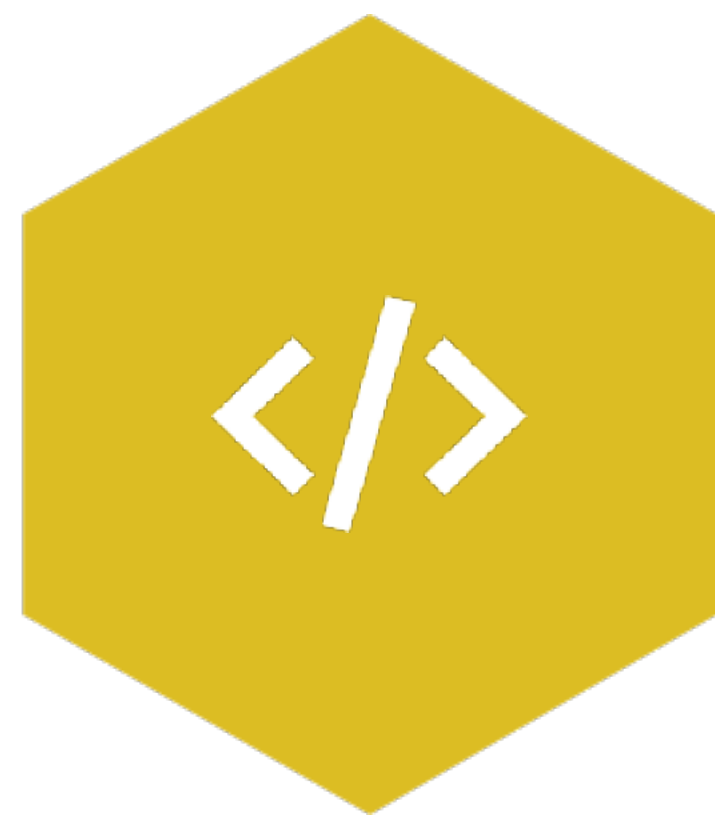


iceworks 3.0

目前日活用户数在 **600+** 左右，主要使用版本是 2.x 和 3.x 并存

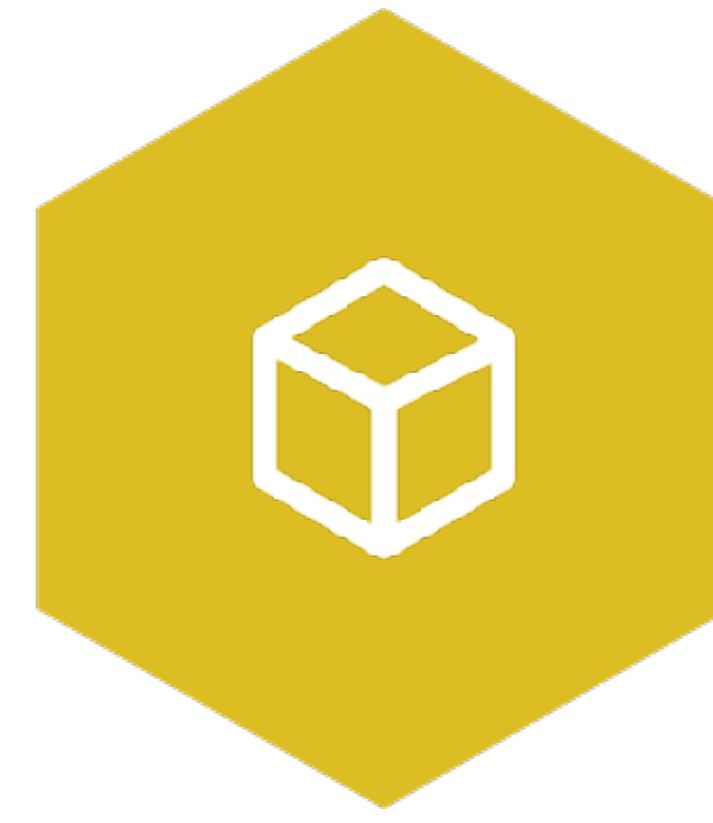
产品结构上主要分为 「项目」、「工程」、「物料」 三大核心能力

产品背景 - 当下问题



1. 编码操作断层

- 和编辑器之间需要频繁切换
- 通过 GUI 操作代码无即视感



2. 提效不彻底

- 新建页面受限于源码物料
- 添加物料操作体验不佳



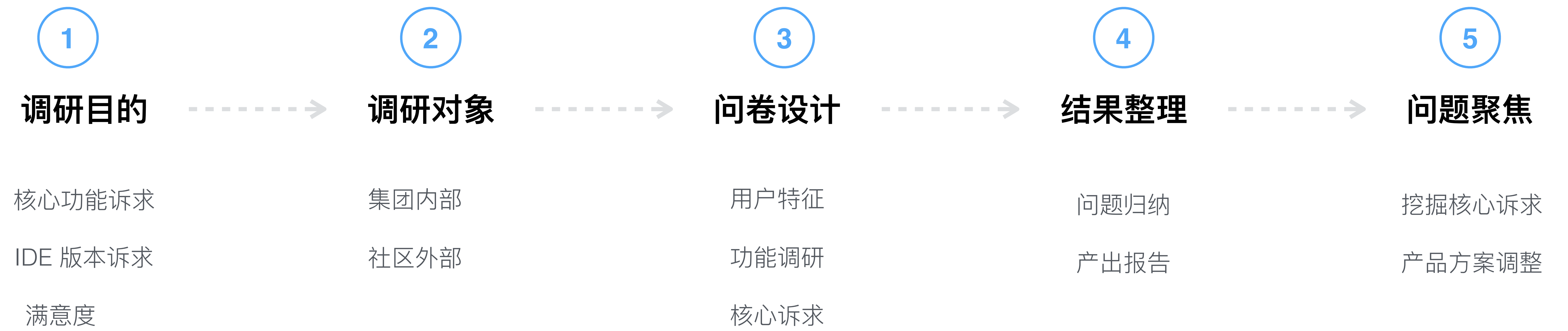
3. 开发链路弱集成

- 简易的 Git / Terminal 等无法满足需求
- 无法覆盖整个开发链路需要切换到其他工具



02. 用户调研

用户调研 - 问卷流程



用户调研 - 问卷设计

1

您的职位是? (单选)

☐

前端

☐

后端

☐

测试

☐

其他

2

您使用的前端框架是? (单选)

☐

React

☐

Vue

☐

其他

3

您在开发工作中遇到的痛点有? (可多选)

☐

页面布局&样式编写繁琐

☐

交互行为&状态管理编写复杂

☐

前后端沟通联调成本高

☐

技术升级迭代快, 方案选择多, 技术门槛变高

☐

开发环境复杂, 经常要折腾前端环境、项目依赖、调试、打包等问题

☐

其他

4

您在使用 iceworks 进行开发吗? (单选)

☐

在使用 iceworks 2.x 版本

☐

在使用 iceworks 3.x 版本

☐

知道 iceworks 但从未使用过

☐

不知道 iceworks

5

您在使用 iceworks 过程中使用最多的 3 个功能是? (可多选)

☐

创建项目

☐

启动项目

☐

构建项目

☐

新建页面

☐

添加区块

☐

路由配置

6

如果 iceworks 提供前端中后台领域的专用 IDE 您会使用吗? (单选)

☐

会, 专用的 IDE 可以提供更专业和更好的开发体验

☐

可能会, 要看提供了哪些有用的功能

☐

不会, 我现在用的 IDE 挺好的, 不希望转变使用习惯

7

您希望 iceworks 新增以下哪些功能? (可多选)

☐

支持通过可视化的方式配置组件的属性、样式和事件

☐

支持通过可视化的方式管理应用的数据源、状态等

☐

支持通过可视化拖拽的方式搭建成区块

8

对 iceworks 有哪些建议或者想法?

请输入你的回答

9

您对 iceworks 的满意度评价是? (单选)

☐

不好用

☐

很一般

☐

还行

☐

好用

10

您的联系方式: 钉钉或者手机号码 (可选, 用于工作人员联系赠送礼品)

请输入你的回答

产品的用户特征

产品的功能使用情况

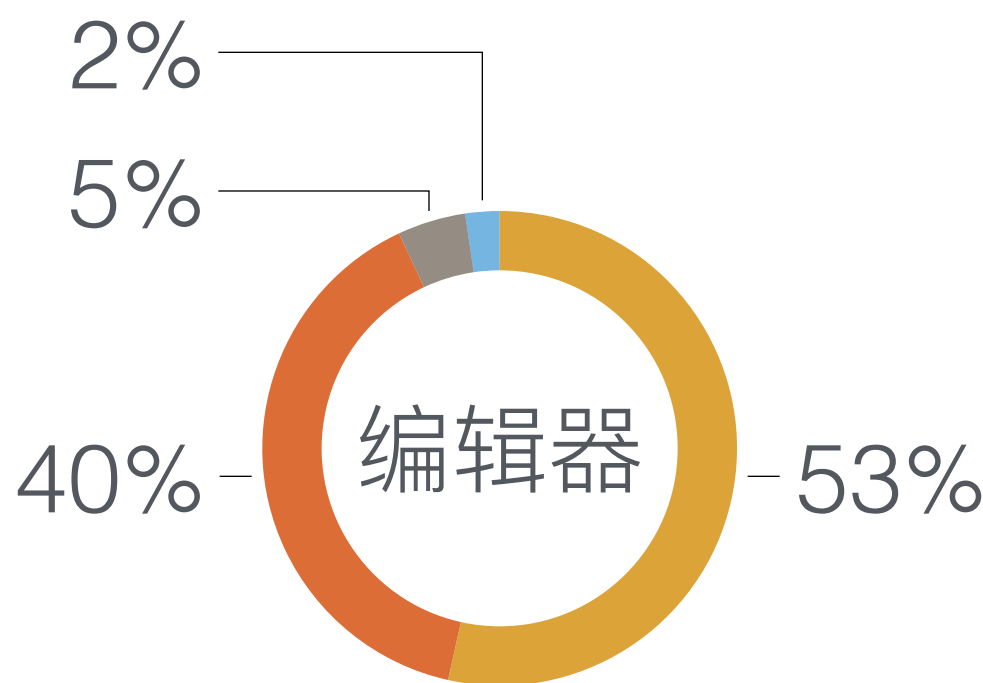
满意度和开放性问题

用户调研 - 有效问卷



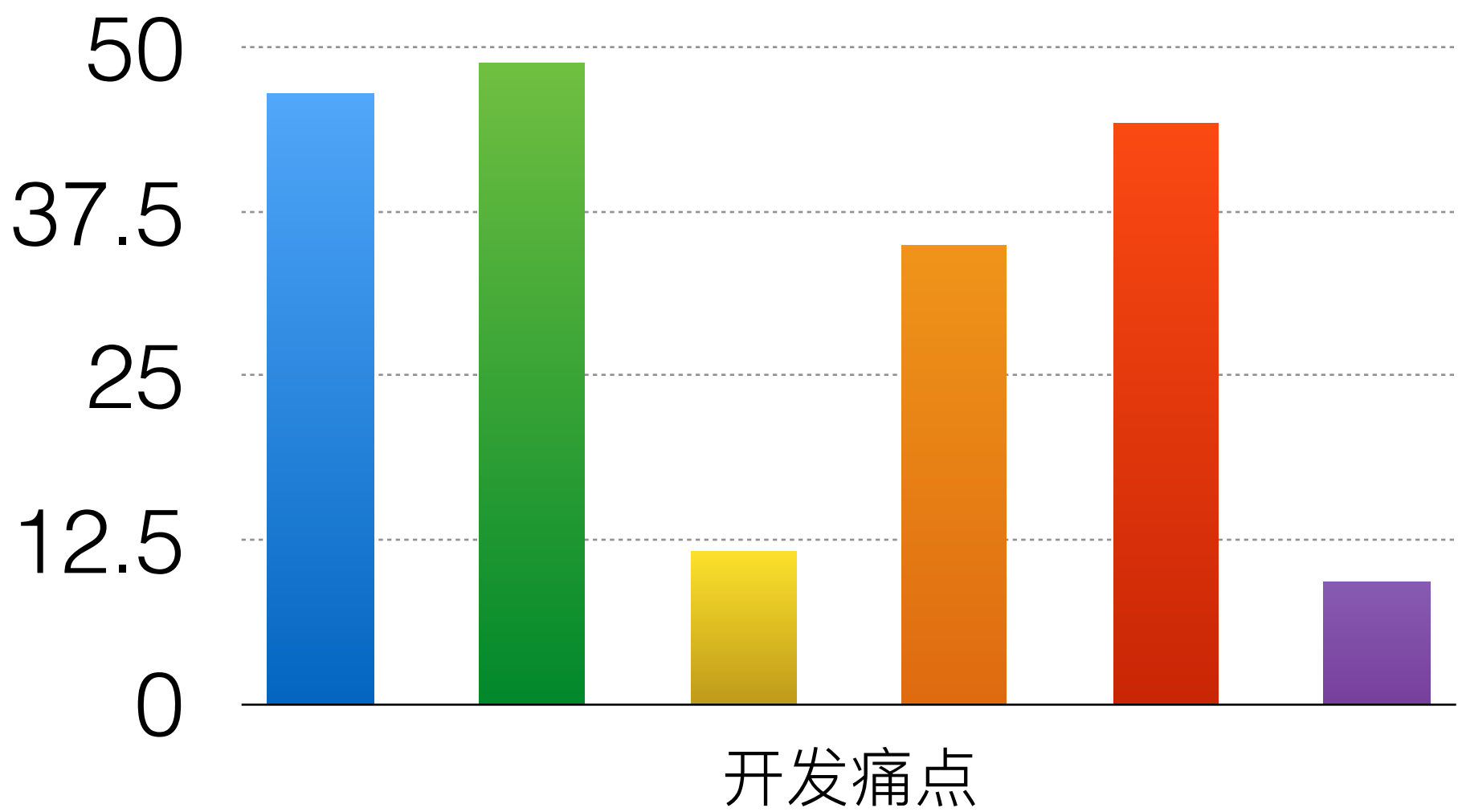
用户调研 - 用户特征

内部



● VSCode: 23 ● WebStorm: 17 ● IntelliJ IDEA: 2 ● 其他: 2

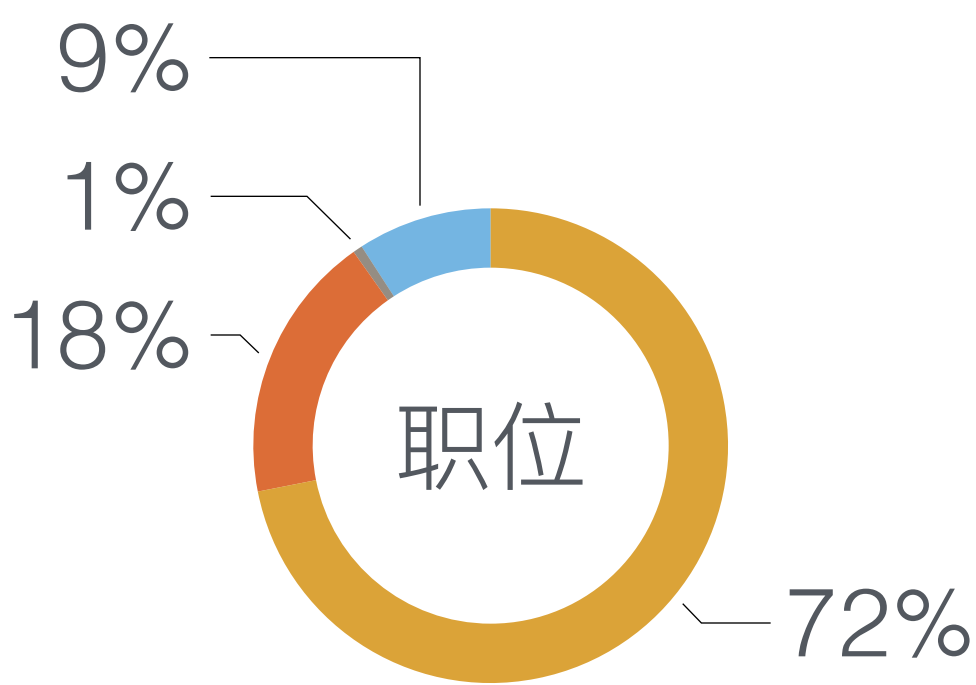
编辑器使用：主要以 VSCode 为主



■ 页面 & 样式编写繁琐: 20 ■ 交互行为 & 状态管理复杂: 21
■ 前后端联调成本高: 5 ■ 技术升级迭代快 & 门槛高: 15
■ 开发环境复杂: 19 ■ 其他: 4

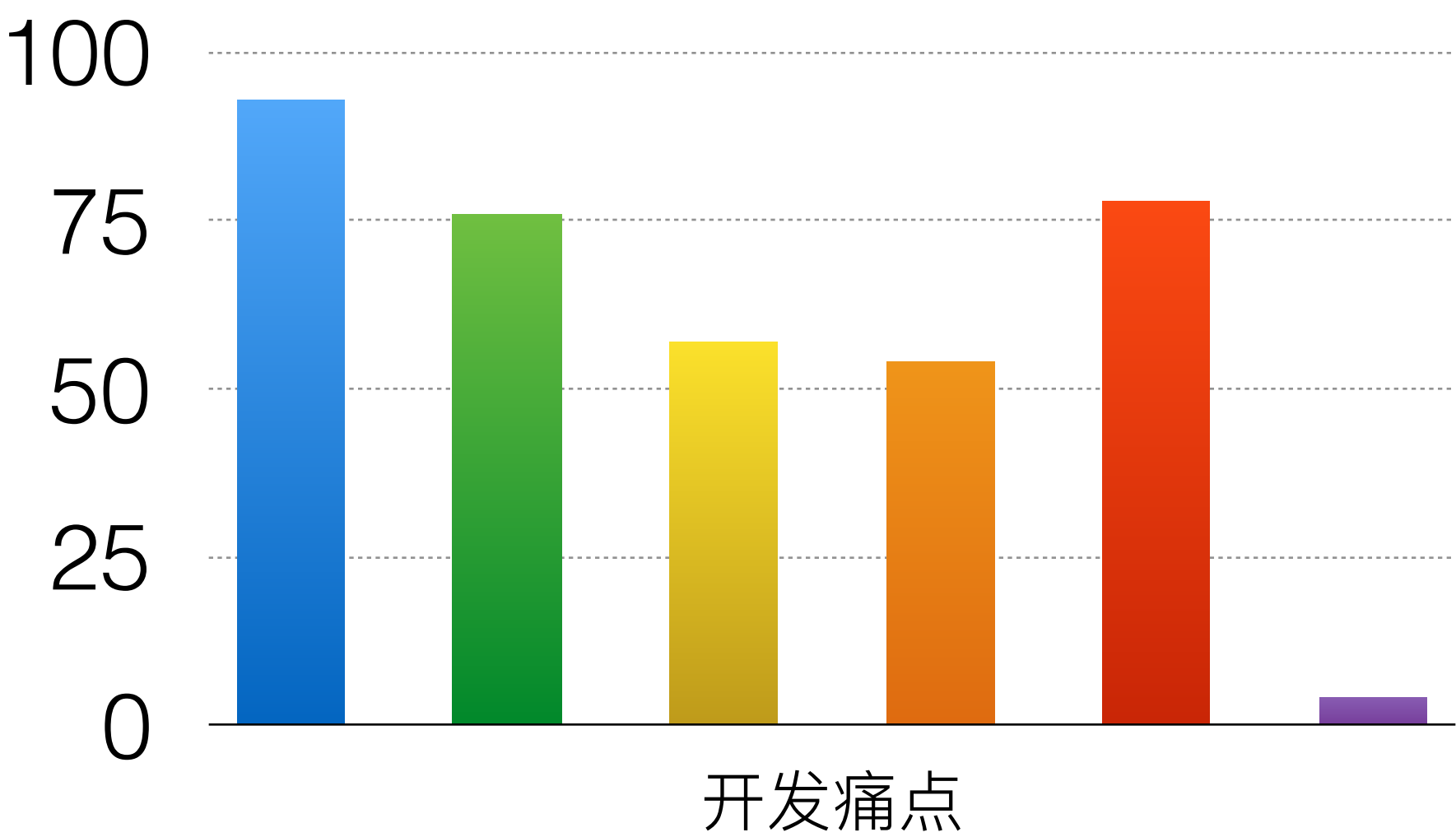
开发痛点：1. 页面 & 样式编写；2. 交互 & 状态管理；3. 开发环境复杂

外部



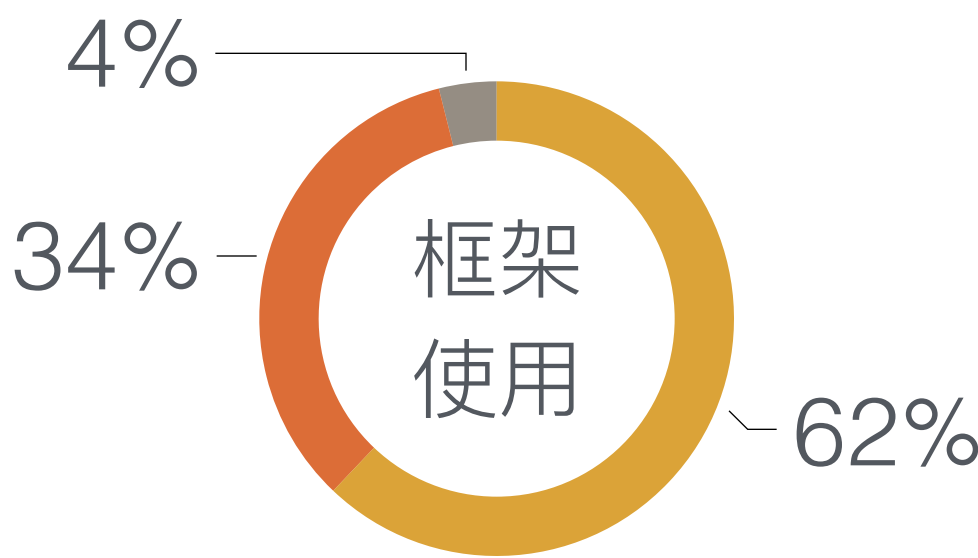
● 前端 ● 后端 ● 测试 ● 其他

用户群体：前端



■ 页面 & 样式编写繁琐: 93 ■ 交互行为 & 状态管理复杂: 76
■ 前后端联调成本高: 57 ■ 技术升级迭代快 & 门槛高: 54
■ 开发环境复杂: 78 ■ 其他: 4

开发痛点：1. 页面 & 样式编写；2. 交互 & 状态管理；3. 前后端联调

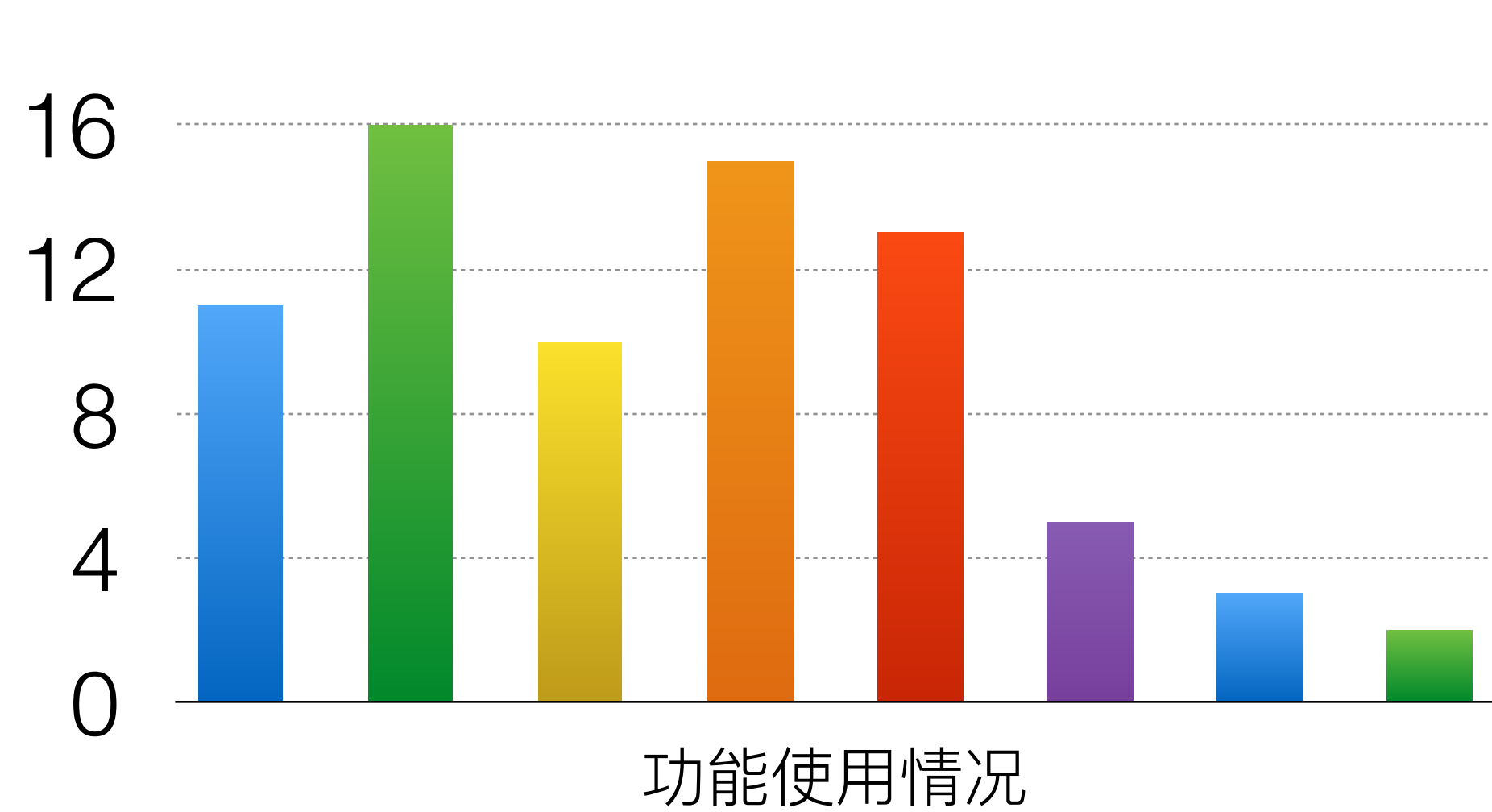


● React ● Vue ● 其他

框架使用最多是：React

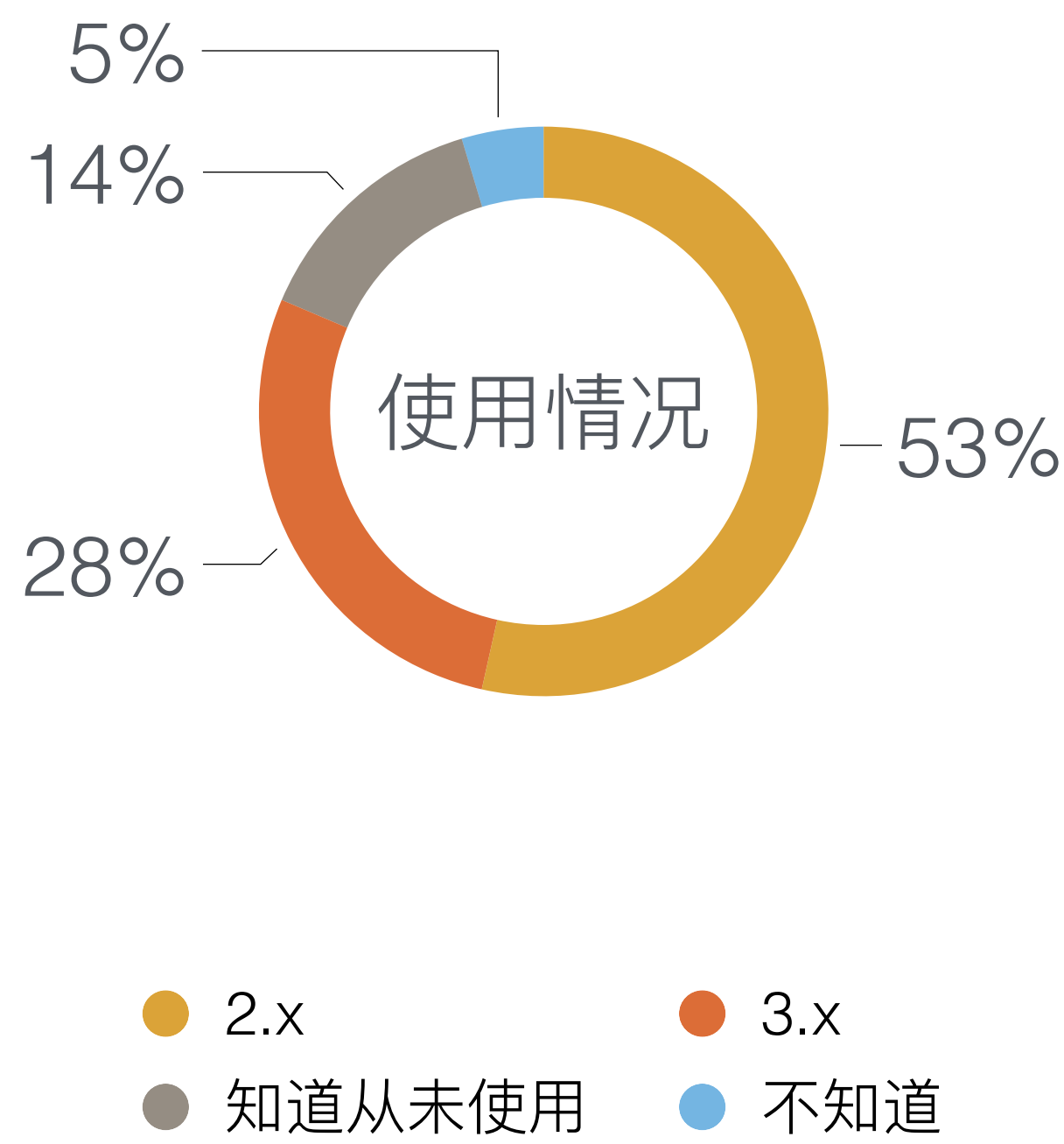
用户调研 - 功能分析

内部

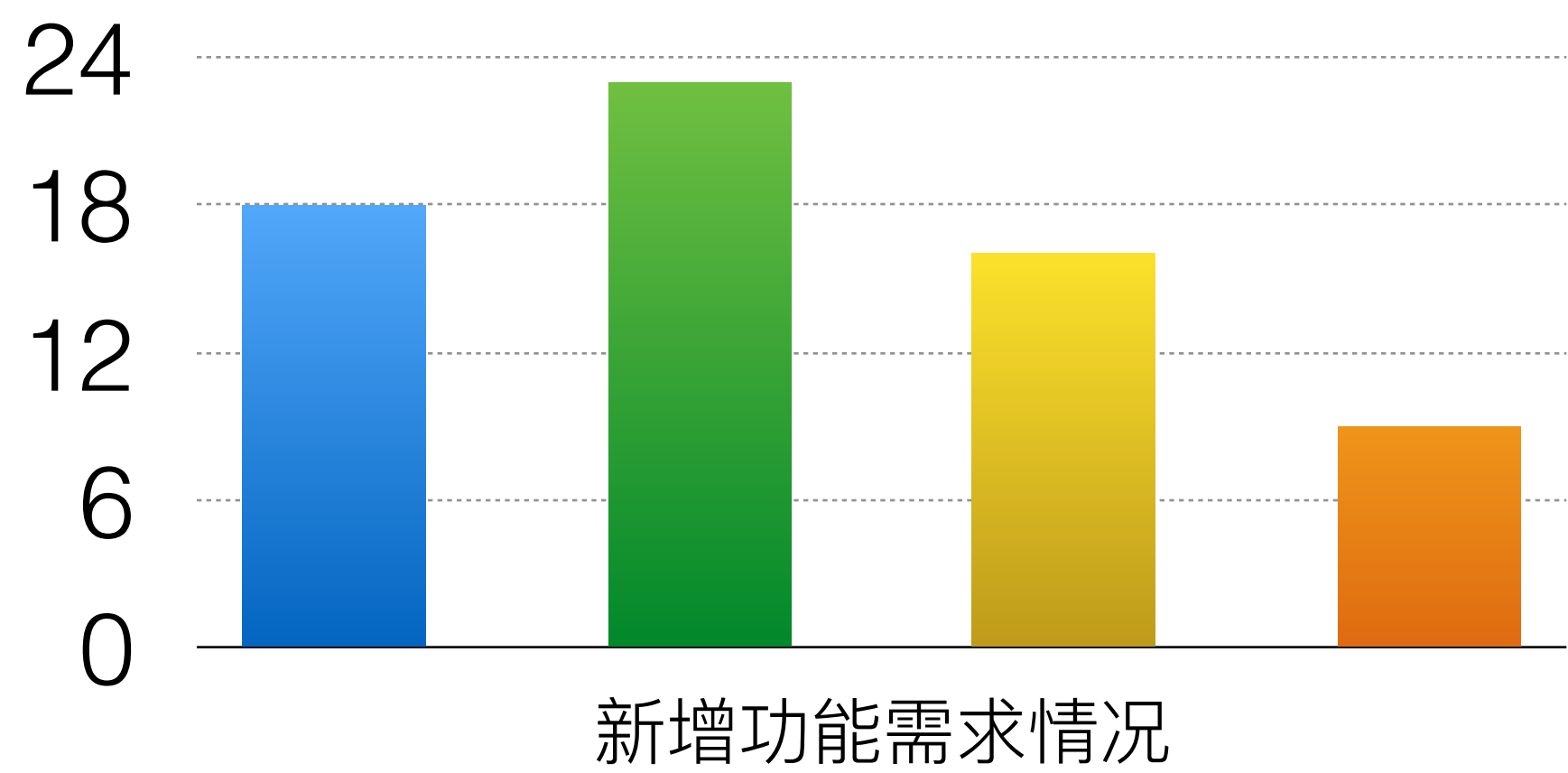


创建项目：11 启动项目：16 构建项目：10
新建页面：15 添加区块：13 路由配置：5
菜单配置：3 其他：2

功能使用前三：1. 启动项目；2.新建页面；3.添加区块



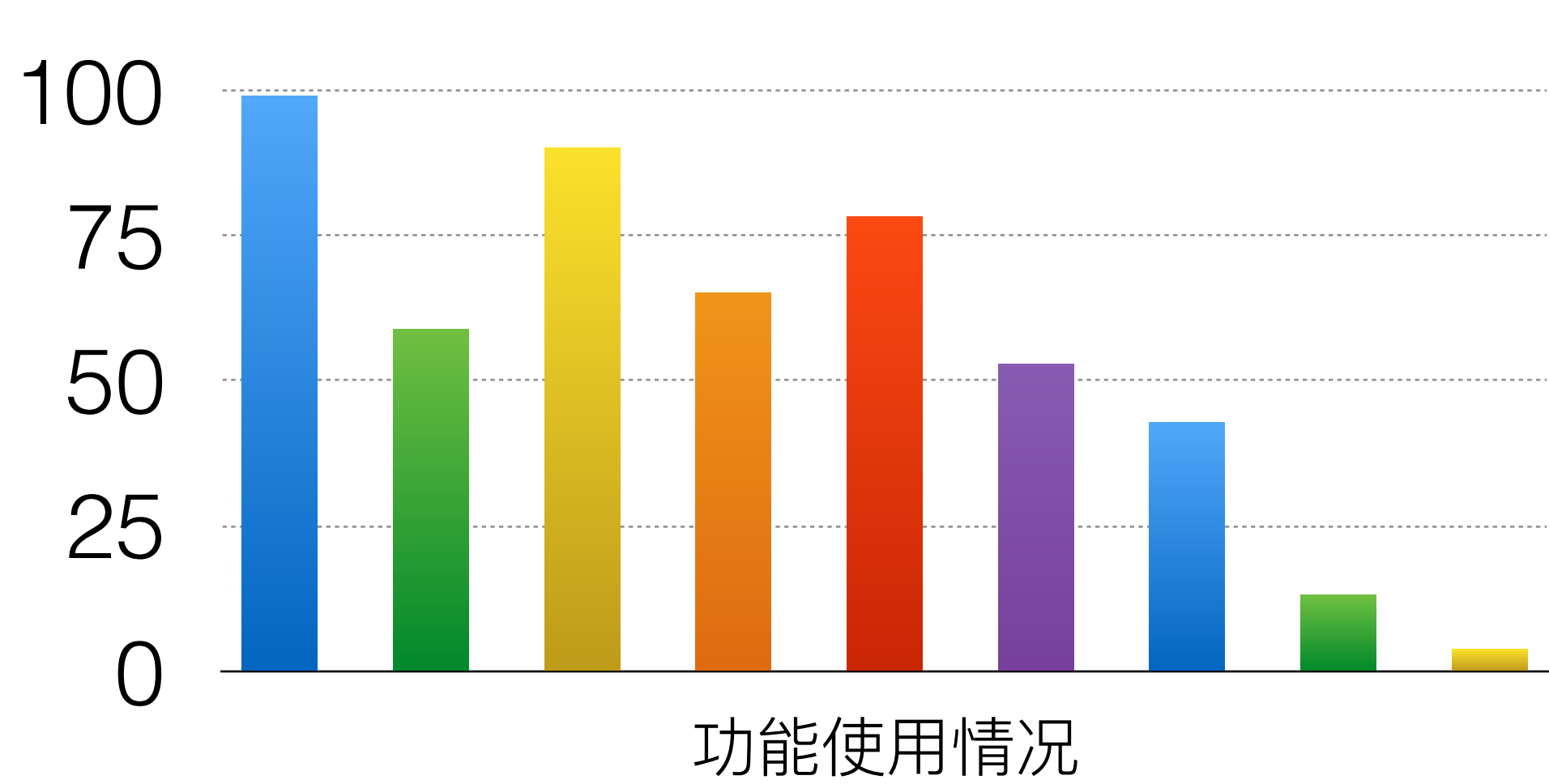
版本使用情况：主要使用 2.x



支持通过可视化的方式配置组件的属性、样式和事件：18
支持通过可视化的方式管理应用的数据源、状态等：23
支持通过可视化拖拽的方式搭建生成区块：16
其他：9

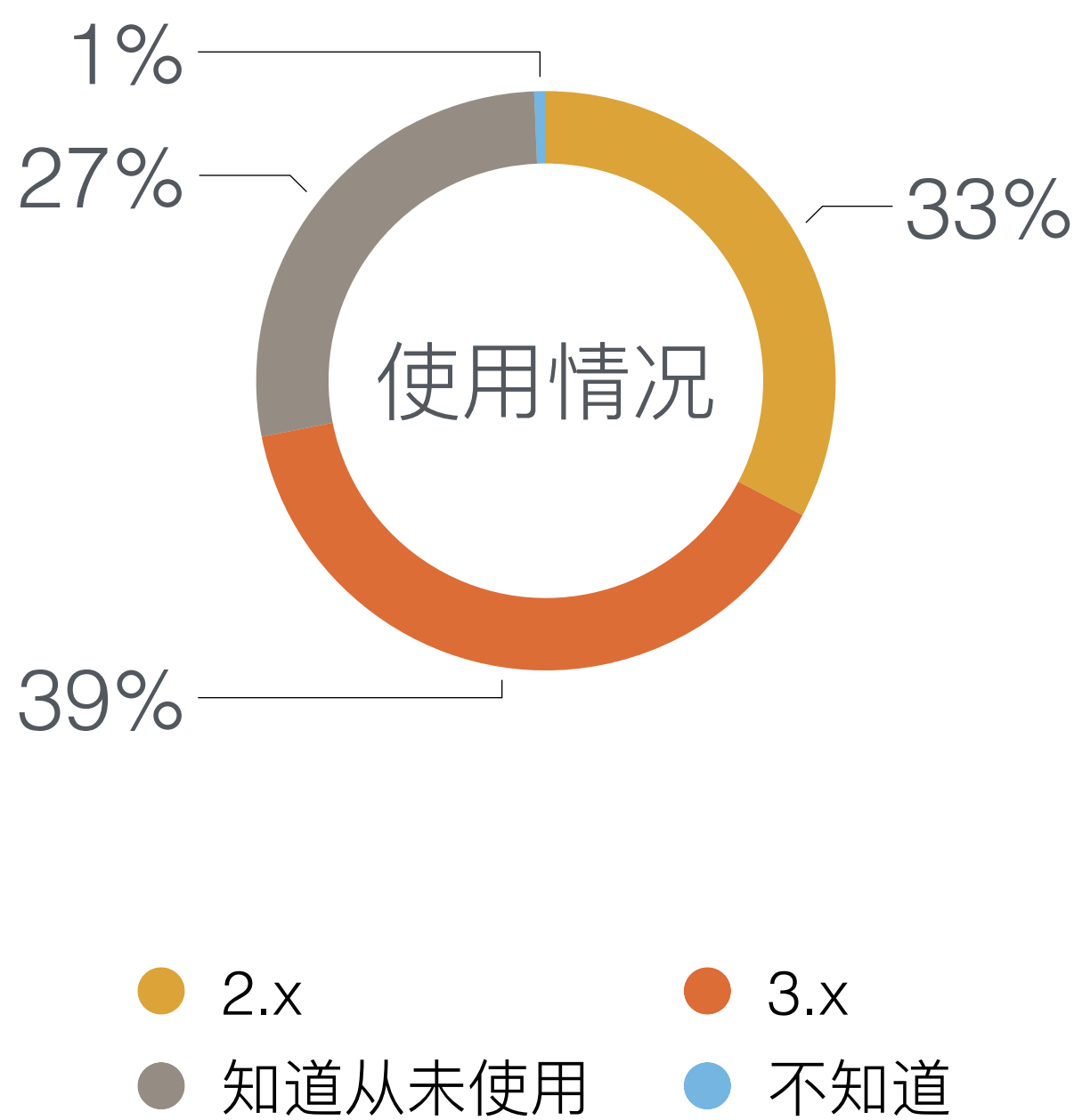
新增功能情况：三种情况几乎均等

外部

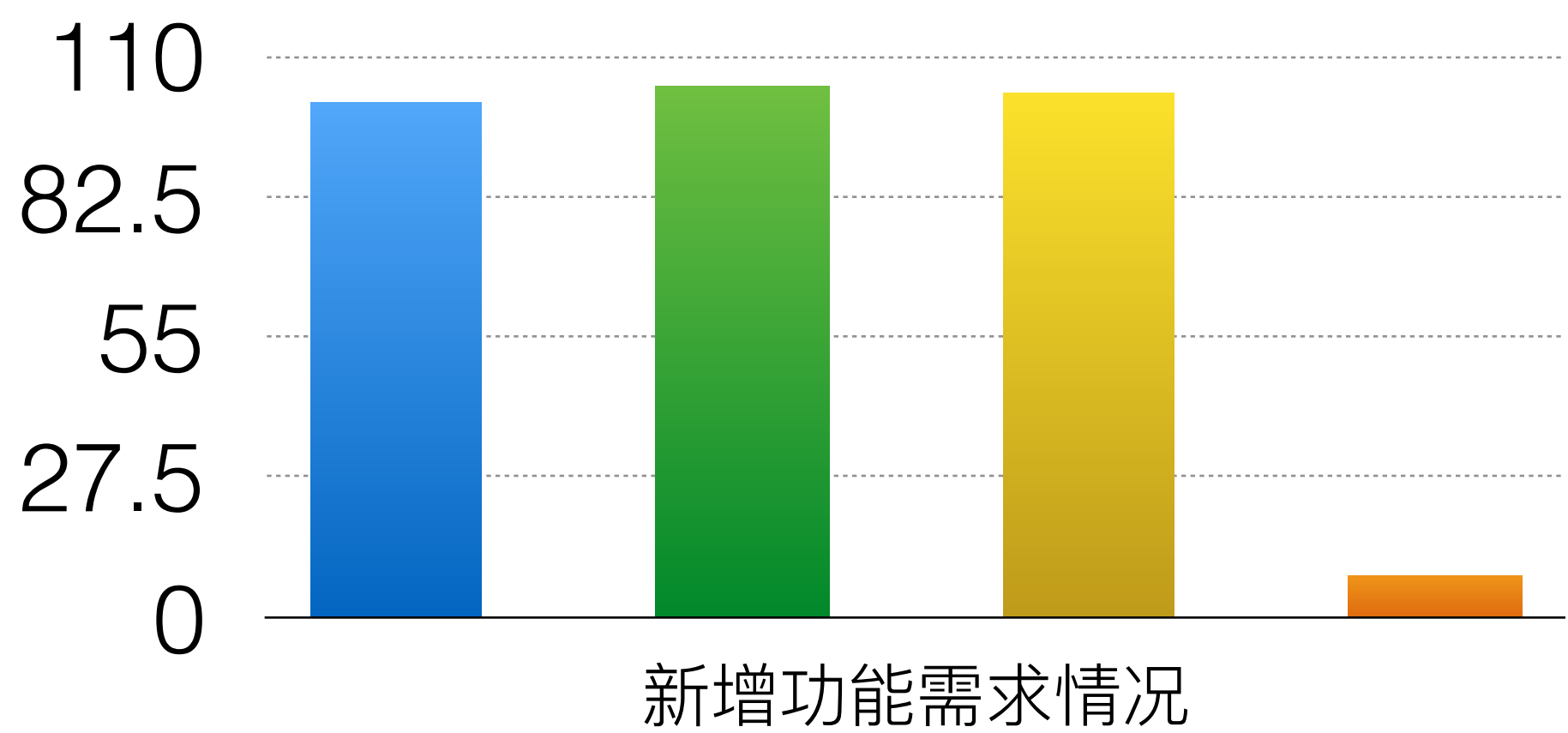


创建项目：99 启动项目：59 构建项目：90
新建页面：65 添加区块：76 路由配置：53
菜单配置：43 其他：13 其他：4

功能使用前三：1. 创建项目；2.构建项目；3.添加区块



版本使用情况：3.x 偏多但相差不大

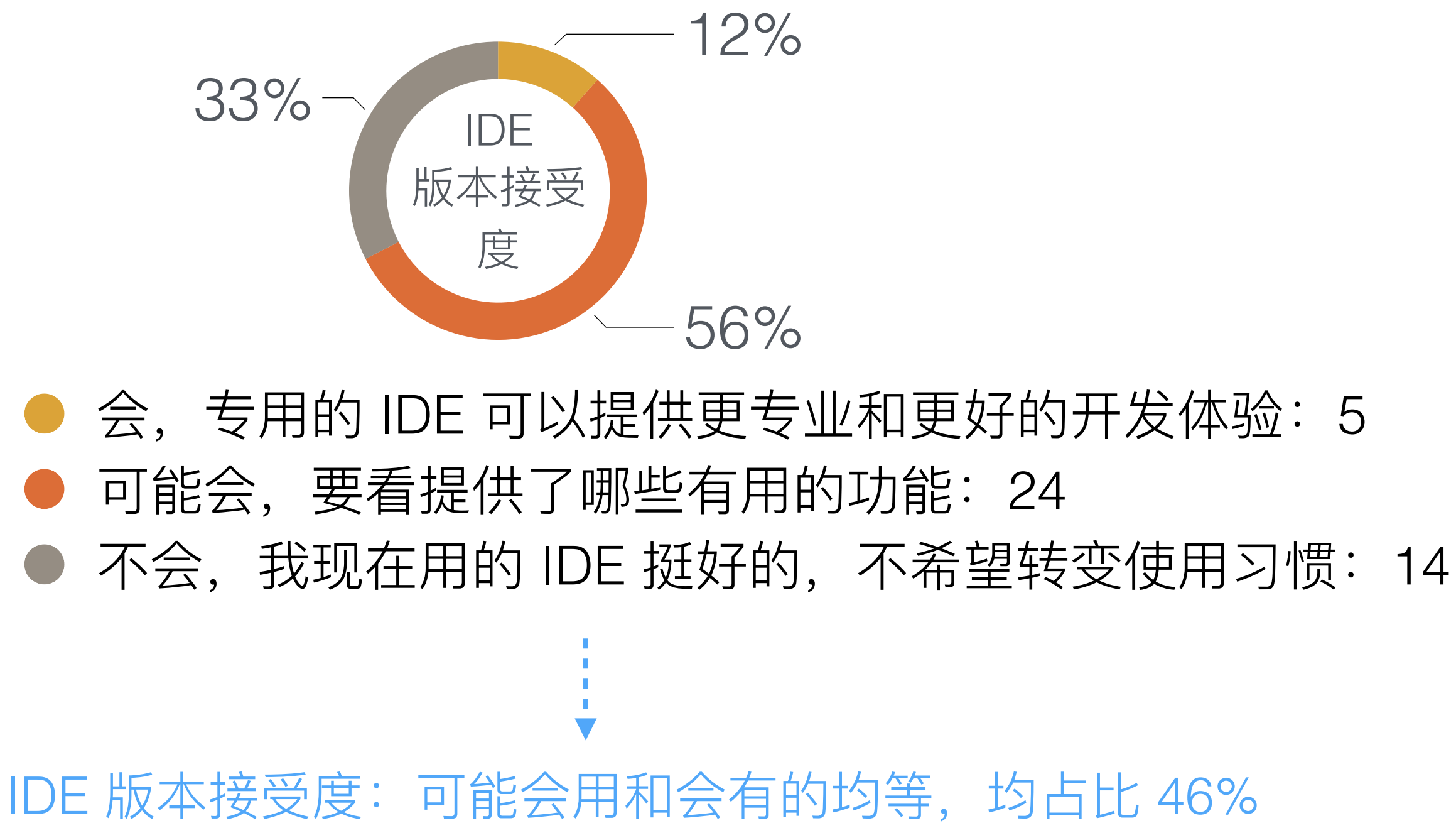
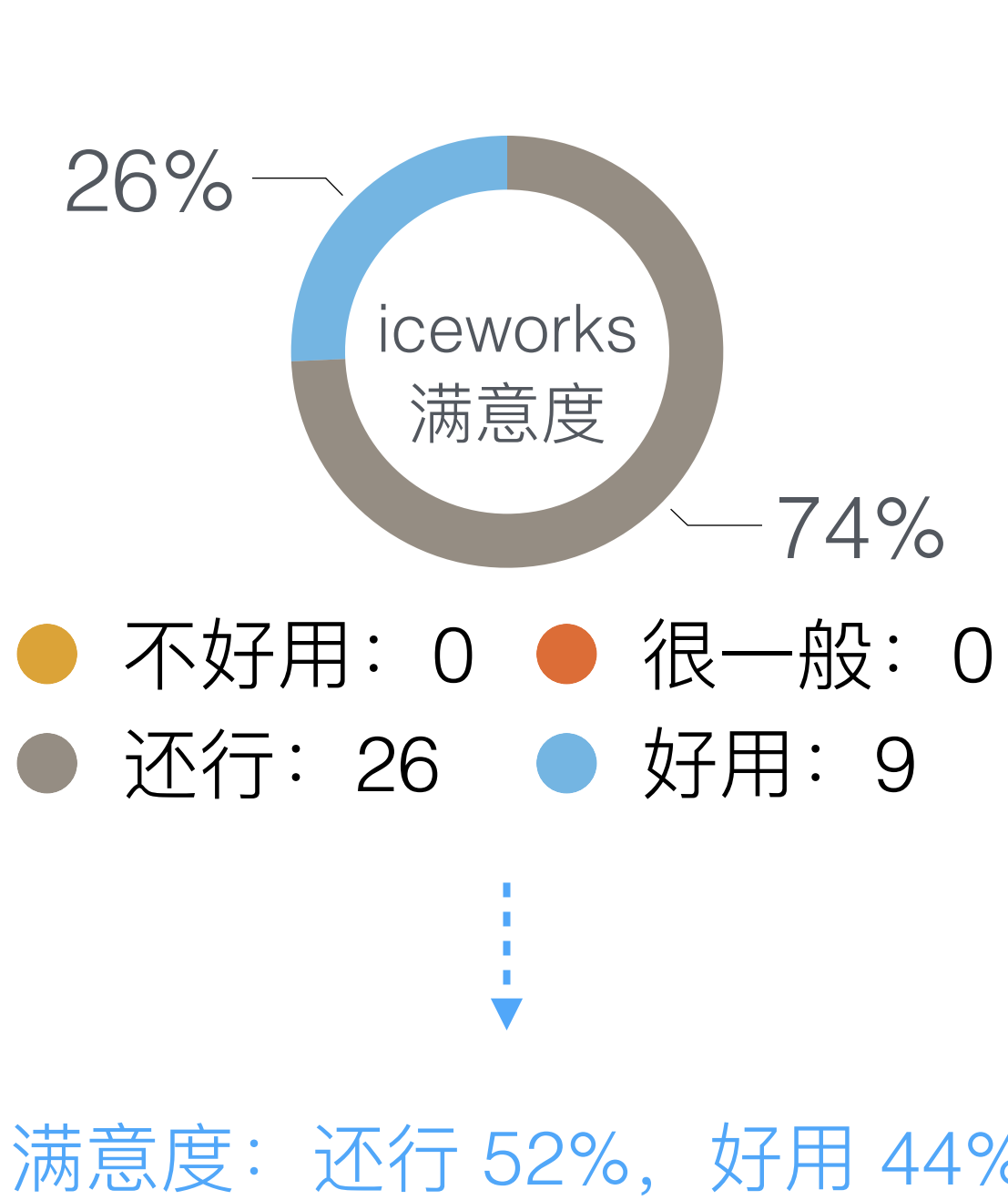


支持通过可视化的方式配置组件的属性、样式和事件：101
支持通过可视化的方式管理应用的数据源、状态等：104
支持通过可视化拖拽的方式搭建生成区块：103
其他：4

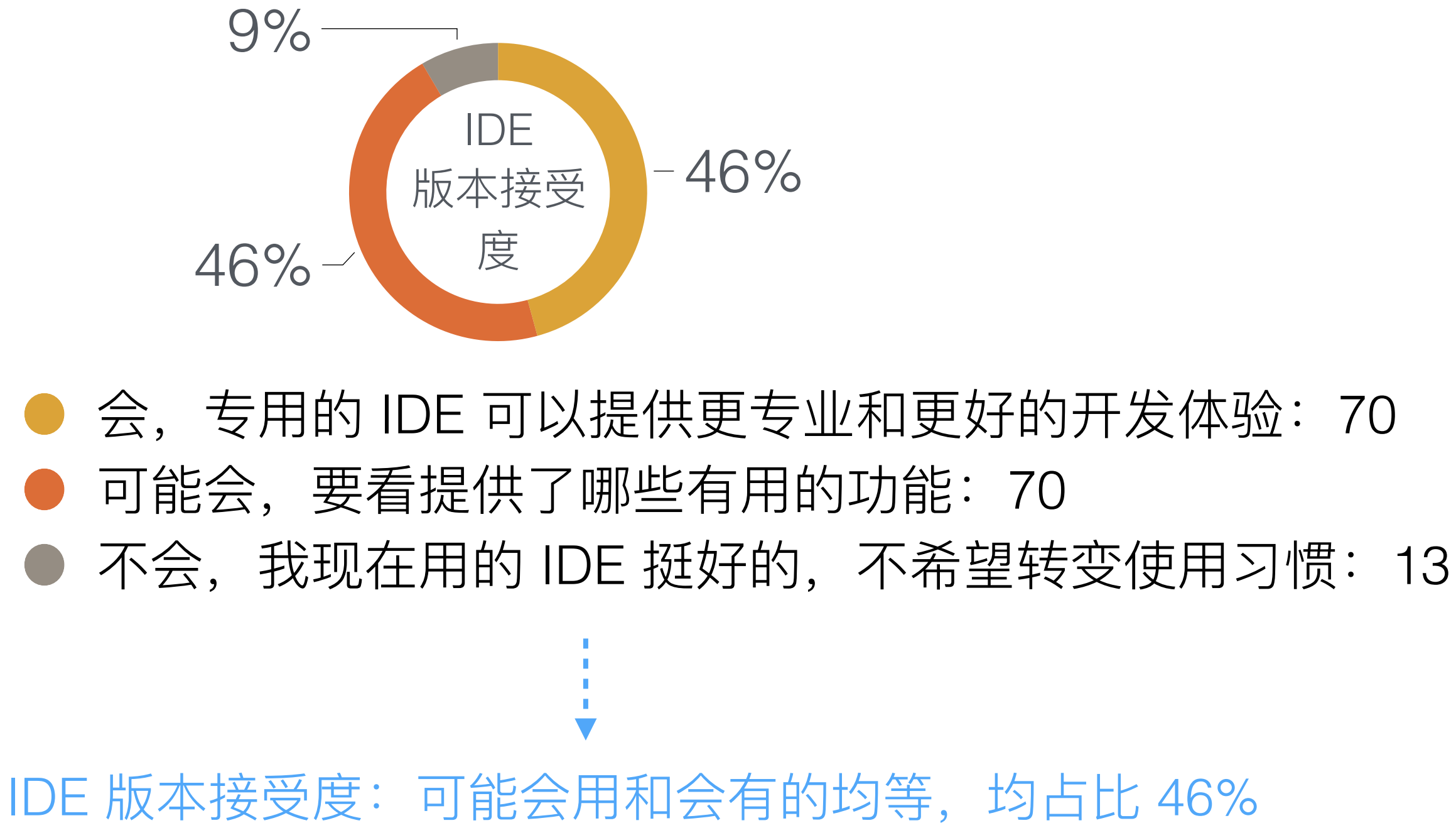
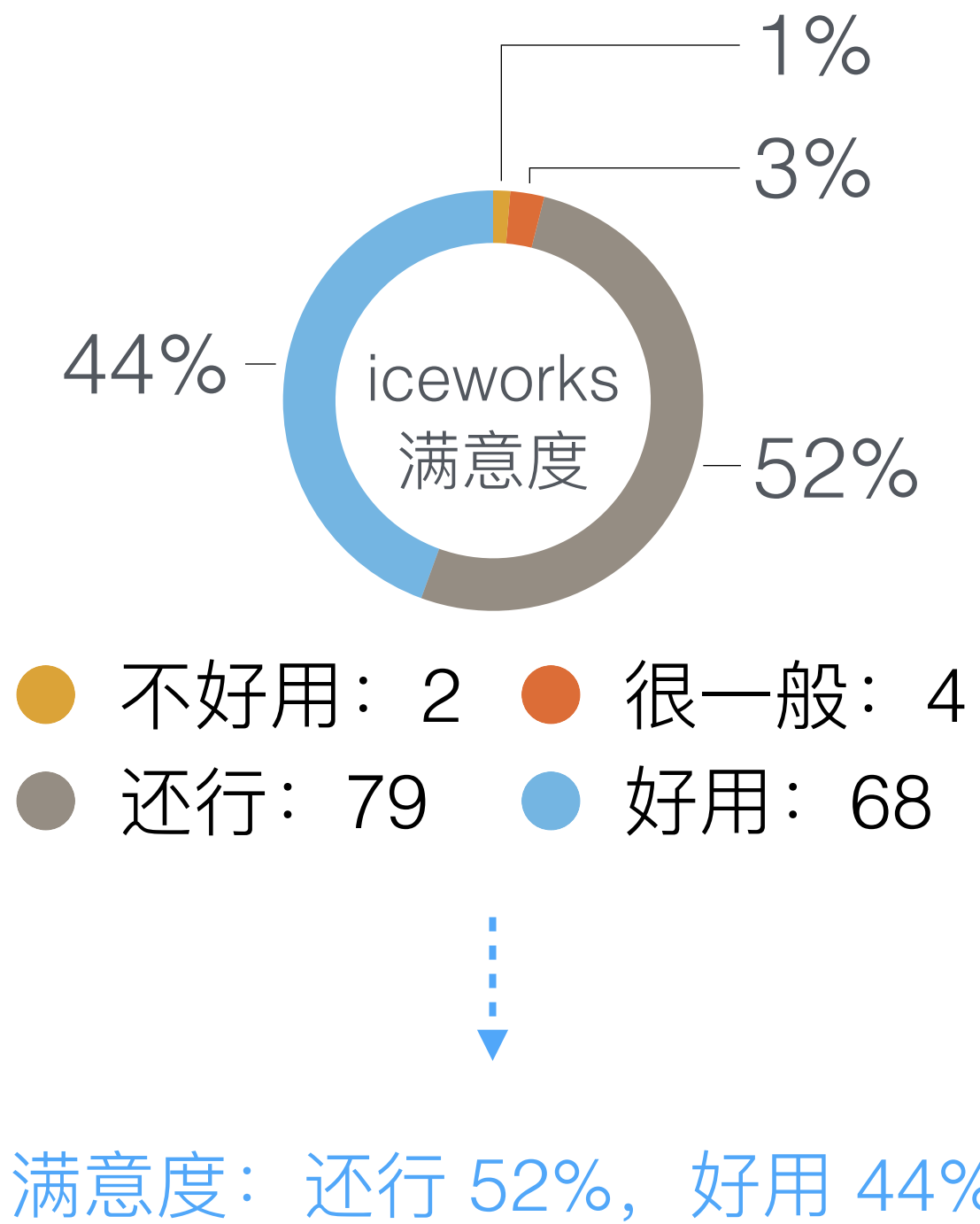
新增功能情况：三种情况几乎均等

用户调研 - 满意度

内部

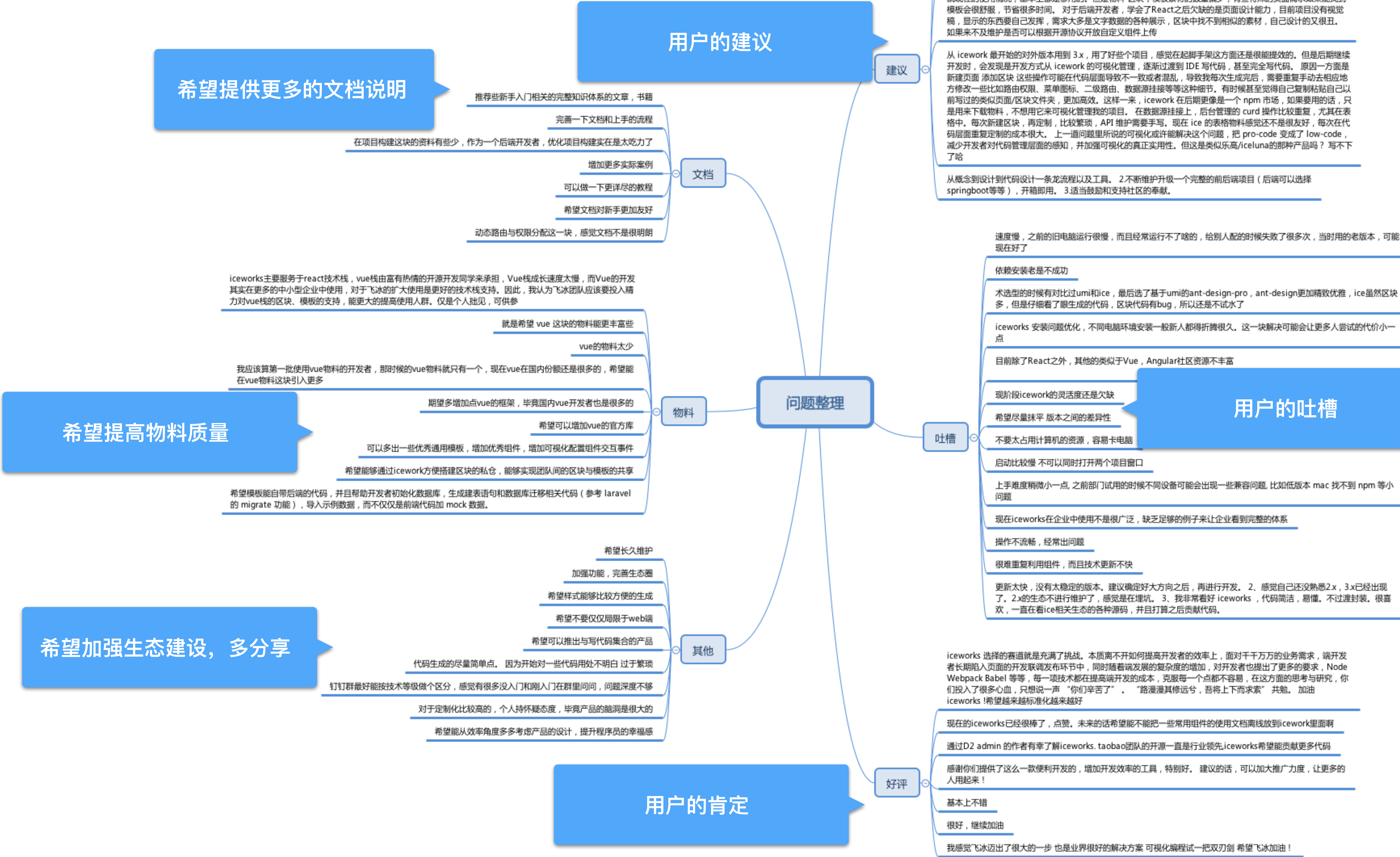


外部



用户调研 - 文本反馈

197份
有效问卷



用户调研 - 文本反馈

用户 A

“飞冰作为前端应用解决方案的集成库，功能是十分强大的，提供了丰富的区块功能。但是就个人来看，飞冰在和其他ui库相比并没有太大的优势，像antd, element-ui等。区块和模板的开发并不能解决项目的开发问题，一个项目的开发最终还是要落到开发者身上，而ui库要做的是怎样减少开发者的工作量，并且提供很好的兼容扩展性能。这就要提供更加丰富的组件功能。飞冰当然已经不错，但是在设计思想上可以更透明化，让使用者更加理解技术本身。

用户 B

“我们还在研究阶段，目前我们使用了物料这块的内容。虽然飞冰目前在开发阶段问题还挺多的，但是感觉这个平台会给我们带来很多期望后续能提供可视化的拖拽区块的功能，可以方便后端人员也可以很快的做布局的事情 期望有些功能对前端是多余的，但是可能对后端是有好的。比如工程操作这些，因为目录结构不一样，所以我们没有使用控制管理平台，因为那样我们要改变很多。目前的飞冰还不能支持控制台的自定义，都只能按着飞冰的走。

用户 C

“对iceworks真诚的一个看法和建议：iceworks的物料虽然不少，但是贴近真实业务的物料太少了，电商领域，最核心和能封装的组件：多规格添加商品，这个难道做不出？iceworks对自己面向的人群有点过于乱了，一方面想要所有人都能通过iceworks来做网站，一方面又不想通过增加特定的业务物料来扩大使用范围，感觉没什么定位，vue和angular都有对应的产品物料，我2018-提出的要有电商场景的物料，谁知道就只有平平无奇的电商后台，iceworks真的是一个好东西，希望能越来越好吧。

197份

有效问卷

number	survey_id	user_id	user_nick	time_star	time_sub	biz	q1.您的职业	q1.您的职业	q2.您使用的技术栈	q2.您使用的技术栈	q3_opt1.	q3_opt2.	q3_opt3.	q3_opt4.	q3_opt5.	q3_opt6.	q3.您在开发中遇到的问题	q4.您在使用的框架	q5_opt1.	q5_opt2.	q5_opt3.	q5_opt4.	q5_opt5.	q5_opt6.
1	38552			2019-09-12 13:11:41	2019-09-12 13:11:41		1.前端		1.React			2.交互行为&状态管理编写复杂						2.在使用iceworks 3.x 版本	1.创建项目					6.路由
2	38552			2019-09-12 10:01:21	2019-09-12 10:01:21		1.前端		1.React			2.交互行为&状态管理编写复杂						1.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	
3	38552		何一和	2019-09-12 13:11:11	2019-09-12 13:11:11		1.前端		1.React						5.开发环境复杂，经常要折腾			1.在使用iceworks 3.x 版本	1.创建项目				5.添加区块	6.路由
4	38552		like宝贝s	2019-09-12 09:48:21	2019-09-12 09:48:21		1.前端		2.Vue			2.交互行为&状态管理编写复杂	5.开发环境复杂，经常要折腾					3.知道iceworks 但从未使用过	1.创建项目	2.启动项	3.构建项目			6.路由
5	38552		杨继昌	2019-09-12 09:39:21	2019-09-12 09:39:21		1.前端		1.React			1.页面布局&样式编写繁琐	技术升级	5.开发环境复杂，经常要折腾				1.在使用iceworks 3.x 版本	1.创建项目	2.启动项	3.构建项目		4.新建页面	5.添加区块
6	38552			2019-09-12 09:48:21	2019-09-12 09:48:21		1.前端		2.Vue			5.开发环境复杂，经常要折腾						3.知道iceworks 但从未使用过	1.创建项目		3.构建项目			6.路由
7	38552			2019-09-12 09:48:21	2019-09-12 09:48:21		1.前端		2.Vue			2.交互行为&状态管理编写复杂						1.在使用iceworks 3.x 版本	1.创建项目		2.启动项	3.构建项目		6.路由
8	38552		王鹏小	2019-09-12 09:46:31	2019-09-12 09:46:31		1.前端		1.React			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					2.在使用iceworks 3.x 版本	1.创建项目		2.启动项	3.构建项目		6.路由
9	38552			2019-09-12 09:54:31	2019-09-12 09:54:31		2.后端		1.React			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					1.在使用iceworks 3.x 版本	1.创建项目					6.路由
10	38552			2019-09-16 10:44:01	2019-09-16 10:44:01		1.前端		2.Vue			1.页面布局&样式编写繁琐	2.交互行	3.前后端沟通联调成本高				2.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	
11	38552			2019-09-12 13:42:01	2019-09-12 13:42:01		1.前端		1.React			1.页面布局&样式编写繁琐	2.交互行	3.前后端沟通联调成本高				2.在使用iceworks 3.x 版本	1.创建项目			4.新建页面	5.添加区块	
12	38552			2019-09-14 17:59:21	2019-09-14 17:59:21		2.后端		2.Vue			1.页面布局&样式编写繁琐	2.交互行为&状态管理编写复杂					3.知道iceworks 但从未使用过	1.创建项目		3.构建项目	4.新建页面		
13	38552			2019-09-12 09:46:41	2019-09-12 09:46:41		1.前端		1.React			1.页面布局&样式编写繁琐	2.交互行	3.前后端	4.技术升级	5.开发环境复杂，经常要折腾		2.在使用iceworks 3.x 版本	1.创建项目	2.启动项	3.构建项目			6.路由
14	38552		苏菲刀客	2019-09-12 17:34:01	2019-09-12 17:34:01		1.前端		1.React					4.技术升级迭代快，方案选择多，技术选型容易				3.知道iceworks 但从未使用过	3.构建项目					
15	38552			2019-09-12 09:55:41	2019-09-12 09:55:41		1.前端		2.Vue			1.页面布局&样式编写繁琐	2.交互行为&状态管理编写复杂	5.开发环境复杂，经常要折腾				1.在使用iceworks 3.x 版本	1.创建项目	2.启动项	3.构建项目	4.新建页面	5.添加区块	6.路由
16	38552			2019-09-12 09:47:51	2019-09-12 09:47:51		1.前端		2.Vue			1.页面布局&样式编写繁琐	2.交互行	3.前后端	4.技术升级	5.开发环境复杂，经常要折腾		1.在使用iceworks 3.x 版本				4.新建页面	5.添加区块	
17	38552			2019-09-16 09:15:41	2019-09-16 09:15:41		1.前端		1.React							6.其他		3.知道iceworks 但从未使用过						
18	38552		农有宝	2019-09-12 09:59:31	2019-09-12 09:59:31		1.前端		1.React					3.前后端沟通联调	5.开发环境复杂，经常要折腾			1.在使用iceworks 3.x 版本					5.添加区块	
19	38552			2019-09-12 10:11:31	2019-09-12 10:11:31		1.前端		2.Vue					4.技术升级迭代快，方案选择多，技术选型容易				1.在使用iceworks 3.x 版本	3.构建项目	4.新建页面	5.添加区块			
20	38552			2019-09-12 10:11:31	2019-09-12 10:11:31		1.前端		2.Vue					4.技术升级迭代快，方案选择多，技术选型容易				1.在使用iceworks 3.x 版本	3.构建项目	4.新建页面	5.添加区块			
21	38552		zhangjia	2019-09-12 19:44:41	2019-09-12 19:44:41		1.前端		3.其他	Angular		1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					2.在使用iceworks 3.x 版本	1.创建项目		3.构建项目			6.路由
22	38552		阿土尔	2019-09-12 10:14:31	2019-09-12 10:14:31		2.后端		2.Vue			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					3.知道iceworks 但从未使用过	1.创建项目		2.启动项	3.构建项目		6.路由
23	38552			2019-09-12 10:14:31	2019-09-12 10:14:31		2.后端		2.Vue			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					3.知道iceworks 但从未使用过	1.创建项目		2.启动项	3.构建项目		6.路由
24	38552		叶雷	2019-09-12 10:51:01	2019-09-12 10:51:01		1.前端		1.React			1.页面布局&样式编写繁琐	2.交互行	3.前后端	4.技术升级	5.开发环境复杂，经常要折腾		3.知道iceworks 但从未使用过			4.新建页面	5.添加区块	6.路由	
25	38552			2019-09-12 11:12:31	2019-09-12 11:12:31		1.前端		2.Vue			1.页面布局&样式编写繁琐	2.交互行	3.前后端沟通联调成本高				1.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	
26	38552		fankeke0	2019-09-16 08:36:51	2019-09-16 08:36:51		4.其他	前端+后端	1.React			2.交互行为&状态管理编写复杂	5.开发环境复杂，经常要折腾					2.在使用iceworks 3.x 版本			4.新建页面		6.路由	
27	38552			2019-09-12 12:48:21	2019-09-12 12:48:21		1.前端		1.React			1.页面布局&样式编写繁琐	2.交互行	3.前后端沟通联调	5.开发环境复杂，经常要折腾			1.在使用iceworks 3.x 版本	3.构建项目			5.添加区块	6.路由	
28	38552		lixiaofei1	2019-09-12 09:49:31	2019-09-12 09:49:31		1.前端		1.React					3.前后端沟通联调成本高				2.在使用iceworks 3.x 版本	1.创建项目	2.启动项	3.构建项目	4.新建页面	5.添加区块	6.路由
29	38552			2019-09-12 10:19:41	2019-09-12 10:19:41		1.前端		1.React			2.交互行为&状态管理编写复杂	4.技术升级迭代快，方案选择多，技术选型容易					2.在使用iceworks 3.x 版本	2.启动项	3.构建项目		5.添加区块		
30	38552			2019-09-12 15:30:31	2019-09-12 15:30:31		1.前端		1.React			1.页面布局&样式编写繁琐	2.交互行为&状态管理编写复杂	5.开发环境复杂，经常要折腾				3.知道iceworks 但从未使用过	1.创建项目	2.启动项	3.构建项目			
31	38552			2019-09-12 10:41:51	2019-09-12 10:41:51		2.后端		1.React			1.页面布局&样式编写繁琐	2.交互行为&状态管理编写复杂					3.知道iceworks 但从未使用过	1.创建项目					
32	38552			2019-09-17 09:41:51	2019-09-17 09:41:51		1.前端		2.Vue			2.交互行为&状态管理编写复杂						1.在使用iceworks 3.x 版本	2.启动项	3.构建项目			6.路由	
33	38552		we62888	2019-09-12 09:48:41	2019-09-12 09:48:41		2.后端		1.React			1.页面布局&样式编写繁琐	2.交互行为&状态管理编写复杂	4.技术升级	5.开发环境复杂，经常要折腾			2.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	
34	38552			2019-09-12 09:48:41	2019-09-12 09:48:41		1.前端		1.React			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					1.在使用iceworks 3.x 版本	1.创建项目					6.路由
35	38552		长虹峰	2019-09-12 12:54:01	2019-09-12 12:54:01		1.前端		1.React			1.页面布局&样式编写繁琐						2.在使用iceworks 3.x 版本	1.创建项目			4.新建页面		6.路由
36	38552			2019-09-12 10:14:31	2019-09-12 10:14:31		1.前端		2.Vue			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					1.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	6.路由
37	38552			2019-09-12 10:14:31	2019-09-12 10:14:31		1.前端		2.Vue			1.页面布局&样式编写繁琐	5.开发环境复杂，经常要折腾					1.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	6.路由
38	38552		商家测试	2019-09-12 15:38:11	2019-09-12 15:38:11		1.前端		1.React			1.页面布局&样式编写繁琐	2.交互行	3.前后端沟通联调	5.开发环境复杂，经常要折腾			2.在使用iceworks 3.x 版本	1.创建项目	2.启动项	3.构建项目			
39	38552			2019-09-12 10:14:31	2019-09-12 10:14:31		1.前端		2.Vue			1.页面布局&样式编写繁琐	2.交互行	3.前后端	4.技术升级	5.开发环境复杂，经常要折腾		3.知道iceworks 但从未使用过	1.创建项目	2.启动项	3.构建项目	4.新建页面		
40	38552			2019-09-12 09:59:31	2019-09-12 09:59:31		2.后端		1.React			1.页面布局&样式编写繁琐	2.交互行	3.前后端沟通联调成本高				2.在使用iceworks 3.x 版本	1.创建项目					6.路由
41	38552			2019-09-12 10:25:11	2019-09-12 10:25:11		1.前端		1.React			2.交互行	3.前后端沟通联调	5.开发环境复杂，经常要折腾				3.知道iceworks 但从未使用过	1.创建项目		3.构建项目		5.添加区块	
42	38552		tb869336	2019-09-16 09:50:41	2019-09-16 09:50:41		1.前端		2.Vue			1.页面布局&样式编写繁琐	3.前后端沟通联调	5.开发环境复杂，经常要折腾				2.在使用iceworks 3.x 版本	1.创建项目		3.构建项目		5.添加区块	

用户 A

“飞冰作为前端应用解决方案的集成库，功能是十分强大的，提供了丰富的区块功能。但是就个人来看，飞冰在和其他ui库相比并没有太大的优势，像antd, element-ui等。区块和模板的开发并不能解决项目的开发问题，一个项目的开发最终还是要落到开发者身上，而ui库要做的是怎样减少开发者的工作量，并且提供很好的兼容扩展性能。这就要提供更加丰富的组件功能。飞冰当然已经不错，但是在设计思想上可以更透明化，让使用者更加理解技术本身。

用户 B

“我们还在研究阶段，目前我们使用了物料这块的内容。虽然飞冰目前在开发阶段问题还挺多的，但是感觉这个平台会给我们带来很多期望后续能提供可视化的拖拽区块的功能，可以方便后端人员也可以很快的做布局的事情 期望界面操作可以更好，目前感觉有些功能对前端是多余的，但是可能对后端是有好的。比如工程操作这些，因为目录结构不一样，所以我们没有使用控制管理平台，因为那样我们要改变很多。目前的飞冰还不能支持控制台的自定义，都只能按着飞冰的走。

用户 C

“对iceworks真诚的一个看法和建议：iceworks的物料虽然不少，但是贴近真实业务的物料太少了，电商领域，最核心和能封装的组件：多规格添加商品，这个难道做不出？iceworks对自己面向的人群有点过于乱了，一方面想要所有人都能通过iceworks来做网站，一方面又不想通过增加特定的业务物料来扩大使用范围，感觉没什么定位，vue和angular都有对应的产品物料，我2018-提出的要有电商场景的物料，谁知道就只有平平无奇的电商后台，iceworks真的是个好东西，希望能越来越好把。

用户调研- 结果概览

用户特征

- 前端开发： 72%
- 框架(React)使用： 62%
- 开发痛点：
 - 开发环境复杂 & 门槛高
 - 页面 & 样式编写
 - 交互 & 状态状态

功能使用

- 创建/启动/构建项目
- 添加页面/区块
- 路由/菜单配置

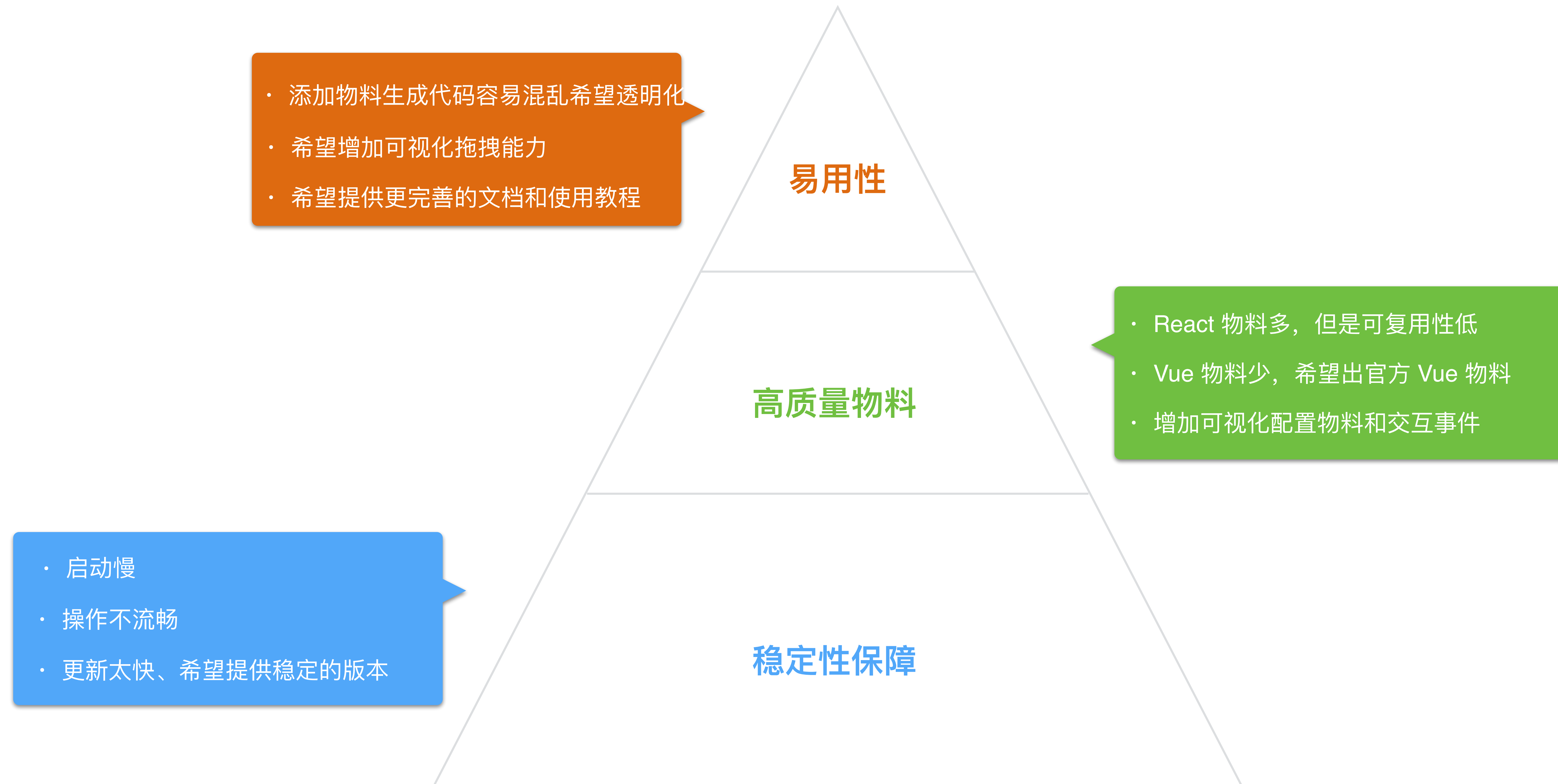
满意度

- 还行： 53.30%
- 好用： 39.08%
- 很一般： 3%
- 不好用： 1%

未来期待

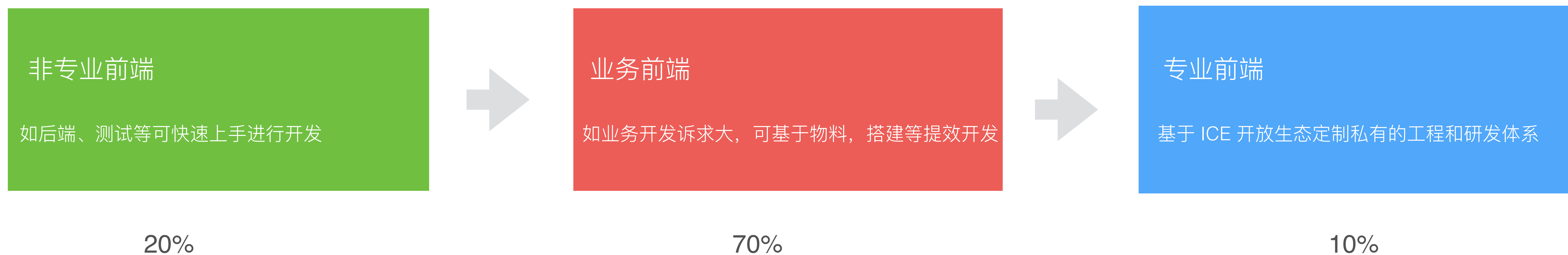
- IDE 版本可接受度： 85.79%
- 高质量物料和可视化搭建
- 加强生态建设和社区输出

用户调研 - 核心诉求



03. 产品规划

目标用户



从调研结果来看，主要目标用户还是前端为主，但主要分为三个层级：「非专业前端」、「业务前端」、「专业前端」

演进趋势

开发前：Node + NPM + CLI 工具 + 编辑器 + xxx

开发时：组件库 + 框架 + Webpack 配置 + 拷贝项目 + 写页面 + xxx

开发完：命令行发布

开发前： iceworks + 编辑器

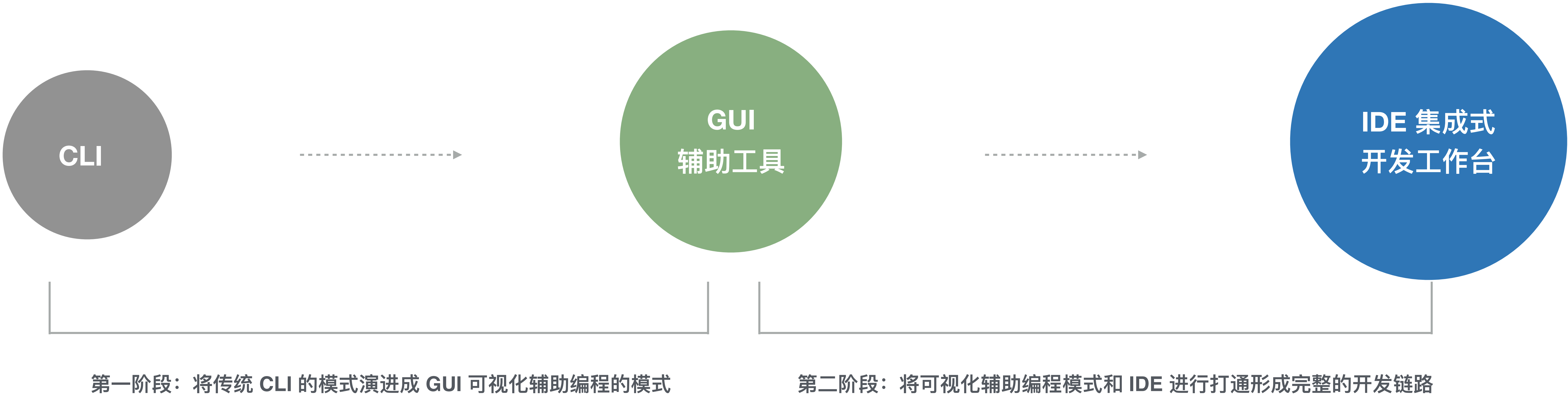
开发时： 可视化创建项目 + 物料组合页面 + xxx

开发完： 一键发布

开发前： 只需要 iceworks

开发时： 可视化创建项目 + 物料组合页面 + 搭建 + xxx

开发完： 一键发布



经过 2 年的发展，可以看到我们的开发模式正在经历比较大的模式变迁，主要分为两个阶段：

1. 第一阶段：核心是将传统 CLI 的模式演进成 GUI 可视化辅助编程的模式，同时基于物料拖拽快速生成页面雏形
同时通过可视化的方式，对本地项目进行启动、依赖管理/构建分析等，整个开发流程可以变得很高效
2. 第二阶段：核心是将 GUI 可视化辅助编程的模式与 IDE 进行打通，形成从编码开始到发布的一个完整开发链路

核心能力

基于物料可视化开发

- 组件 & 区块可视化添加到源码
- 源代码中的组件可视化配置属性
- 物料开发流程可视化

可视化搭建辅助

- 区块搭建
- 组件搭建
- 页面搭建

文件可视化配置

- 框架层配置文件：菜单 & 路由
- 数据 & 状态可视化配置
- 工程和依赖可视化配置

iceworks IDE 集成式开发环境

编码操作断层

- 和编辑器之间需要频繁切换
- 操作无即视感

提效不彻底

- 新建页面只能选择已有物料
- 添加区块操作体验不佳

开发链路弱集成

- 无法覆盖整个开发链路需要切换到其他工具
- 简易的 Git / Terminal 等无法满足需求

产品大图

服务业务

小二工作台

新零售中后台业务

其他中后台业务

产品形态

桌面客户端

工作台

资源管理

项目

列表

创建

导入

删除

物料源

列表

添加

删除

...

编码链路

基于物料可视化开发

组件添加

属性配置

物料开发

...

配置文件可视化设置

菜单

路由

数据

...

可视化搭建辅助

区块搭建

物料搭建

项目搭建

...

工程链路

工程任务

启动调试

执行构建

代码检查

...

依赖管理

增卸升

...

发布链路

DEF

OSS

...

Git

源码开发工作台

可视化搭建工作台

自定义模板工作台

Terminal 终端

代码理解

文件操作

文本搜索

Git 仓库管理

...

基础设施

研发框架标准

路由方案

状态方案

工程方案

...

搭建工作台

画布面板

属性面板

Schema2Code

...

IDE 生态

插件机制

扩展机制

定制机制

...

产品链路



差异竞争力化

基于 IDE 的
一体化开发

通过 IDE 提供开箱即用的
中后台集成式开发环境

源码
可视化配置

源码开发存在一定的学习成本，通过将
将繁杂的属性配置，数据源，状态管理
等进行可视化展示，降低理解和开发成本

混合开发能力

通过源码和可视化搭建融合的方式
提供混开发能力，可以在源码和搭
建之间任意切换

面临挑战



开发者 IDE 工具的使用习惯已经形成



IDE 的稳定性需要得到保障



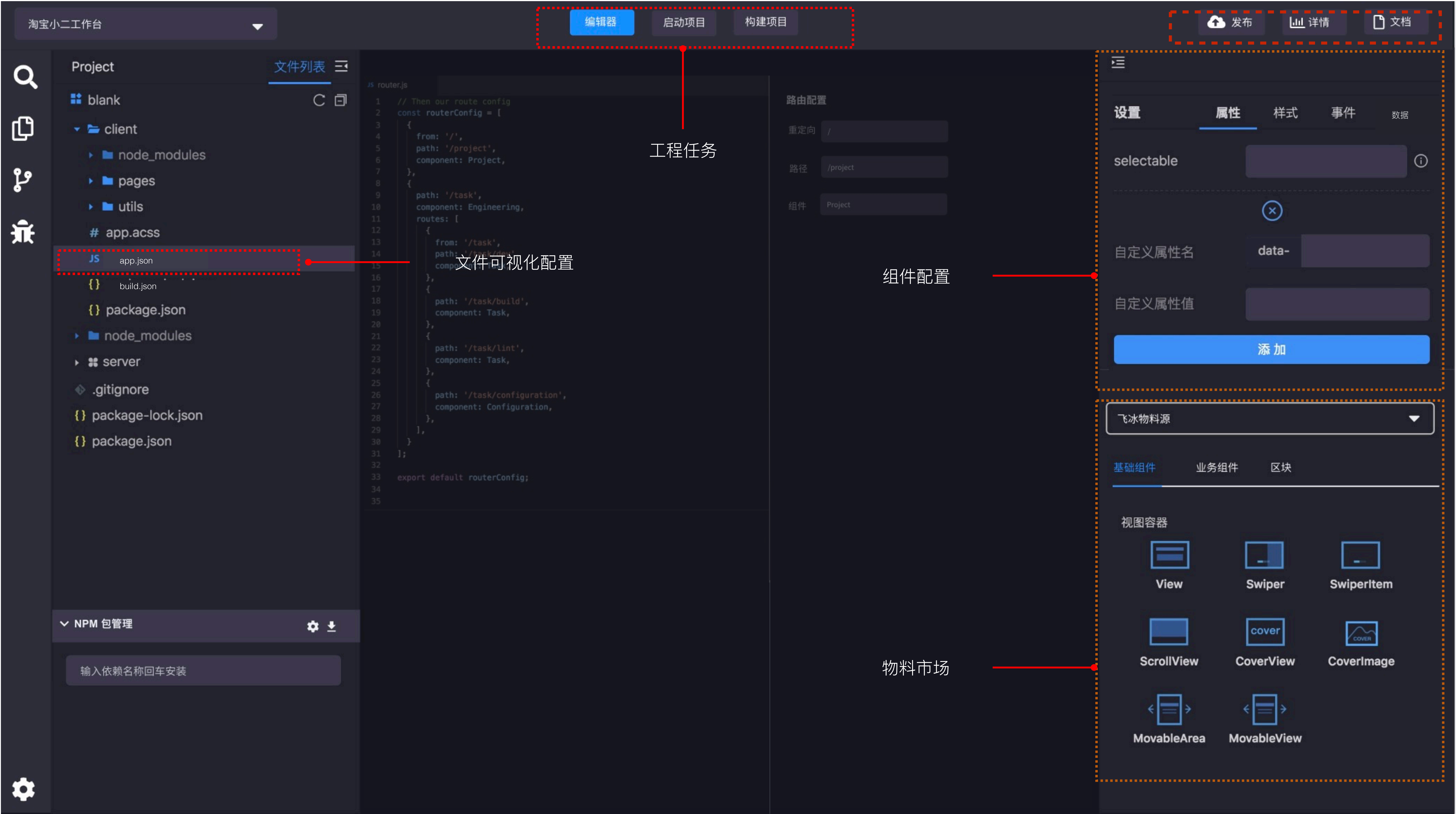
源码与搭建的互转没有成熟的技术方案



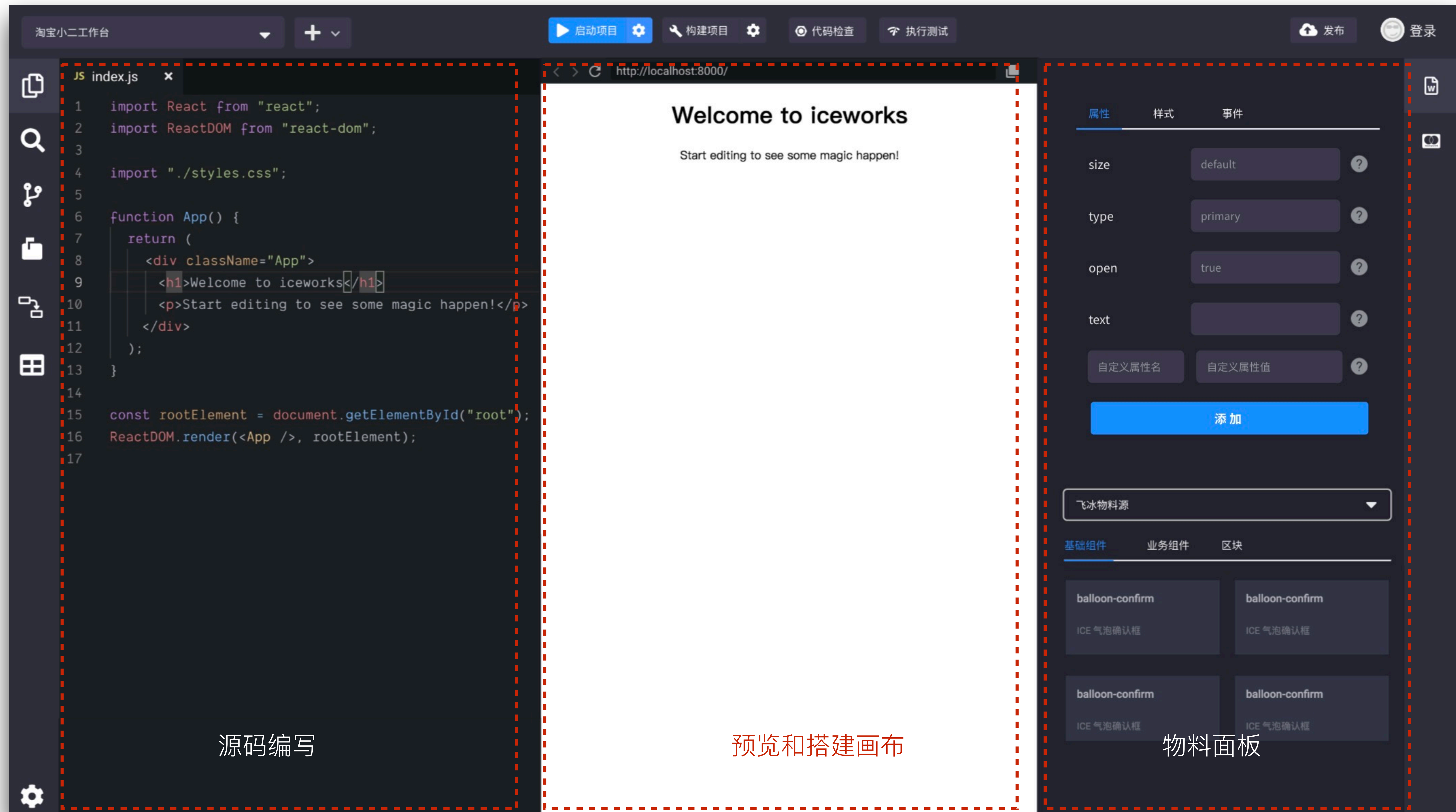
产品

技术

产品原型 - 工作台



产品原型 - 混合开发



衡量数据

物料使用率

- 组件使用率：
- 区块使用率：
- 模板使用率：

文件可视化配置效率提升

- 菜单配置：
- 路由配置：
- 工程配置：

可视化搭建效率提升

- 页面搭建：
- 区块搭建：
- 组件搭建：

产品核心数据： DAU、使用时长、业务覆盖率、社区影响力

THX