CODE A BETTER LIFE
一 行 代 码　亿 万 生 活

如何打造一个高效的组件库

张胜

美团

张胜
美团支付 前端

多年前端研发经历，负责美团支付钱包方向研发及团队基础技术组件建设

Vix 介绍

- Vue 组件库
- 美团金融的标准组件库
- 60+ 个线上业务在使用


Accordion
标题
有研究显示，打字的时候不喜欢在中文和英文之间加空格的人，感情路都走得很辛苦，有七成的比例会在 34 岁的时候跟自己不爱的人结婚，而其余三成的人最后只能把遗产留给自己的猫。半竟爱情跟书写都需要适时地留白。
标题


Banner
1
手动滑动渐隐渐现效果 active: 1
当前位置
1
循环播放
自动播放
手动滑动百分比
0


Button
按钮
按钮
按钮
长按钮
不可用
渐变按钮
不可用渐变按钮
按钮
按钮


BankcardSheet
弹出银行卡列表
选择支付方式
银行卡余额不足，更换银行卡即可完成支付（120113）
中国工商银行(3292)
信息待完善
去完善
中国银行(8911)
中国工商银行(2032)
浦发银行(0578)


BillDetail
充值中
¥63.84
缴费业务
燃气费
缴费单位
北京市燃气集团有限责任公司
客户编号
支付方式
招商银行储蓄卡（9966）
优惠明细
美团优惠￥3.3
银联优惠￥9.8
交易时间
2016-09-26 20:30
关联交易
退款-减脂鸡胸套餐
处理进度
支付成功，处理中
02-06 14:24
还款成功
02-06 14:25
查看退款详情

什么是高效的组件库？

高效组件库的三个特点

A

UI 风格一致

只有 UI 风格与视觉侧保持一致，高还原视觉稿，才能快速完成界面开发。

B

可复用性强

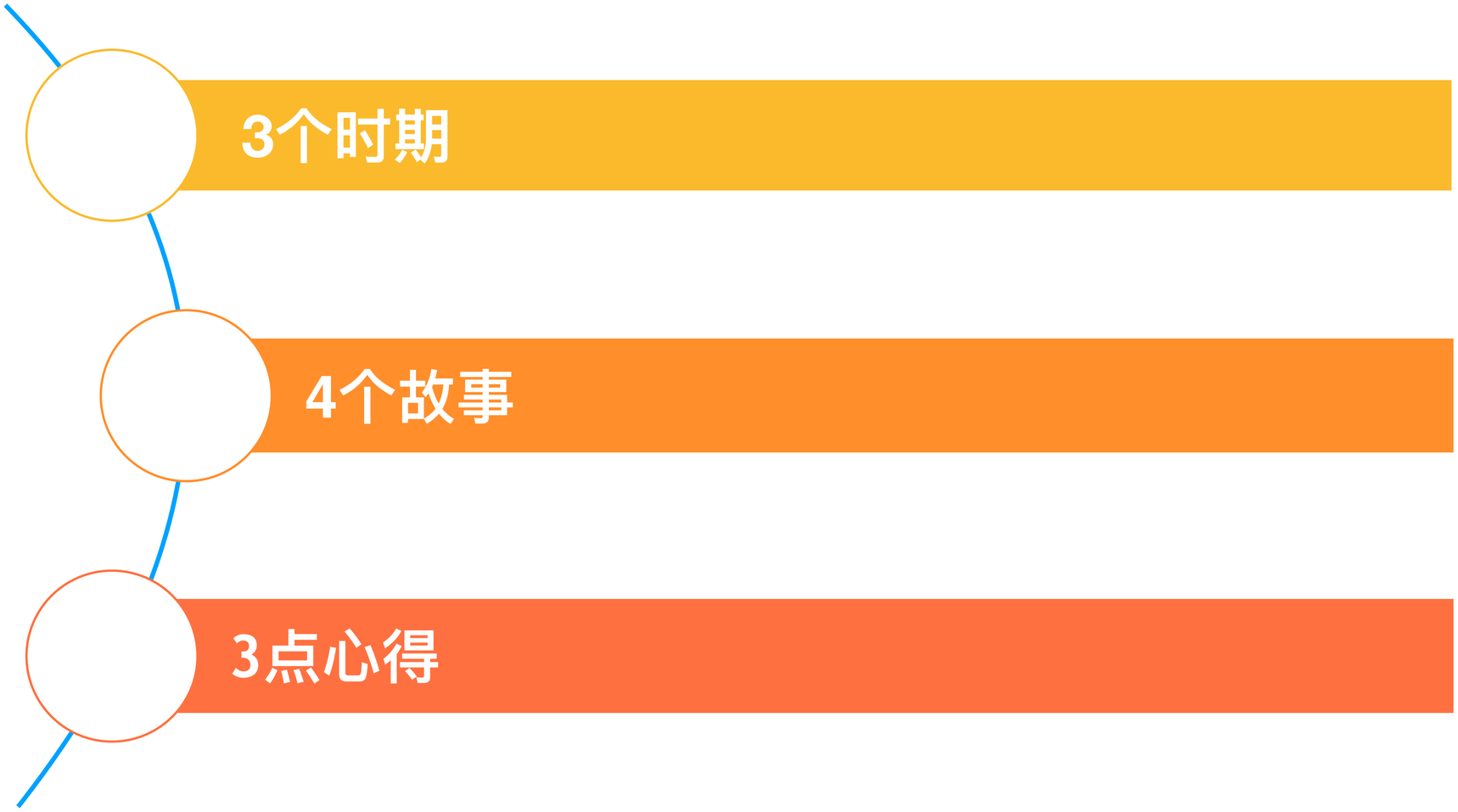
只有可复用性强，才能覆盖更多场景，获得更多收益。

C

简洁易上手

只有简洁易上手，才能让更多人使用，创造更大的价值。

接下来要讲的内容



Vix 创建过程

从产生，到发展，再到推广到各部门使用，有挑战，有汗水，也在一步步创造更大的价值。

2016/02

产生期

做出来。集中爆发，自由贡献。

2017/03

发展期

落地使用。解决问题，大力发展。

2018/01

优化期

做精。耐心打磨，精益求精。

Vix 是如何创建的?

为什么要做组件库？

01 为什么要做组件库？

不想重复开发；不想复制粘贴。

02 为什么团队需要一个组件库？

提高开发效率；保持一致体验；保证代码质量。

03 为什么不使用开源组件库？

UI 不一致；功能不满足；改造成本大。

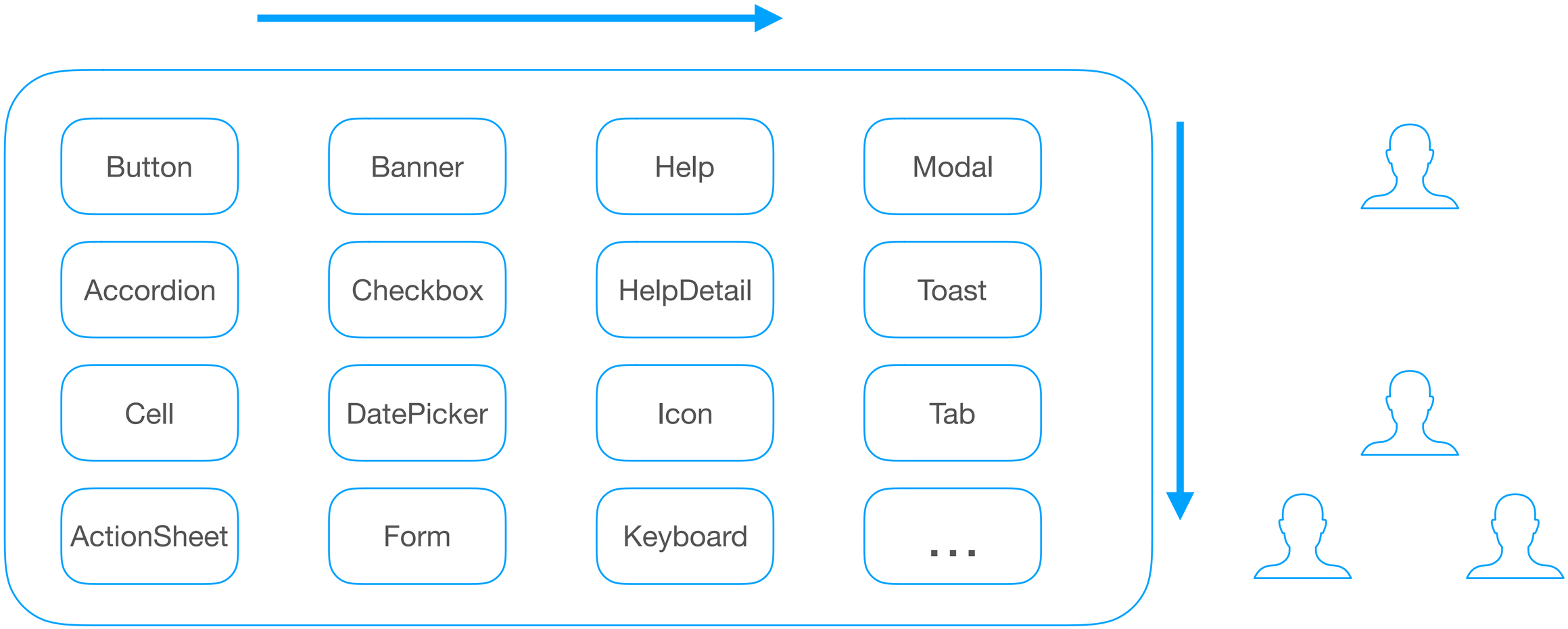
Vix 创建第一个组件

01 明确需求　02 设计接口　03 完成开发

- 按钮可长可短
- 是否有背景色
- 可用、不可用状态

size	normal	large
type	normal	outline
state	normal	disabled

Vix 组件一步步增多


Button
Banner
Help
Modal
Accordion
Checkbox
HelpDetail
Toast
Cell
DatePicker
Icon
Tab
ActionSheet
Form
Keyboard
...

发展期

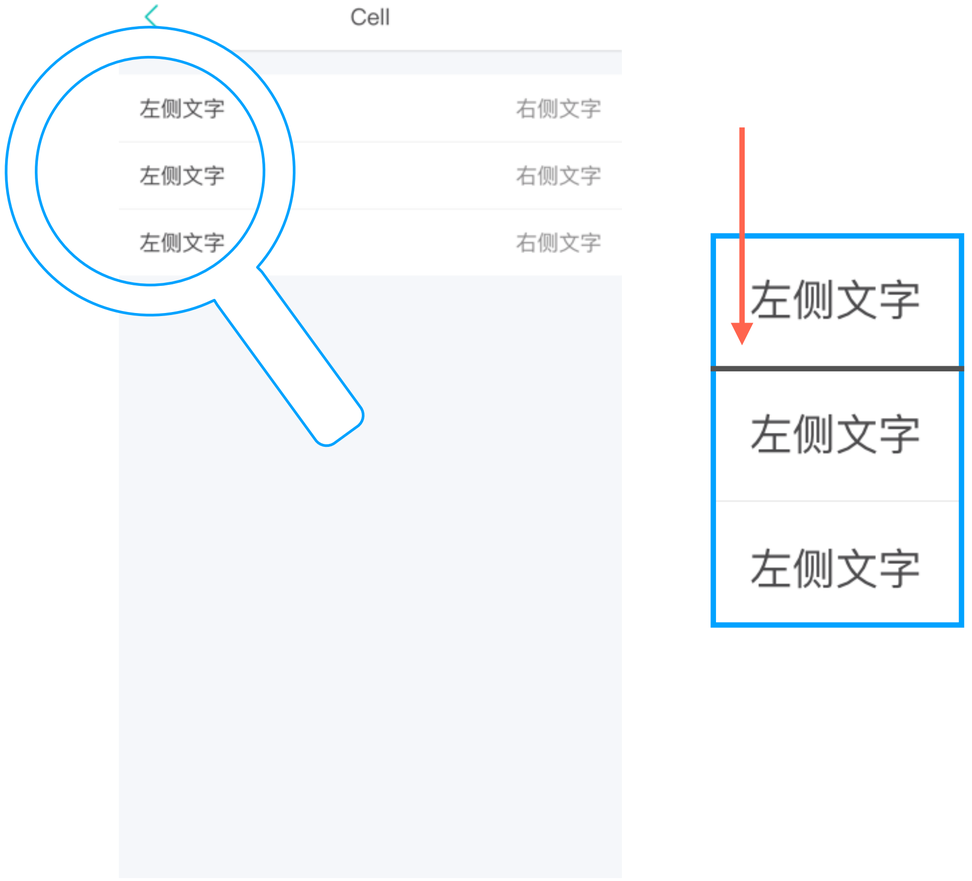
Vix 如何形成一致的 UI 风格?

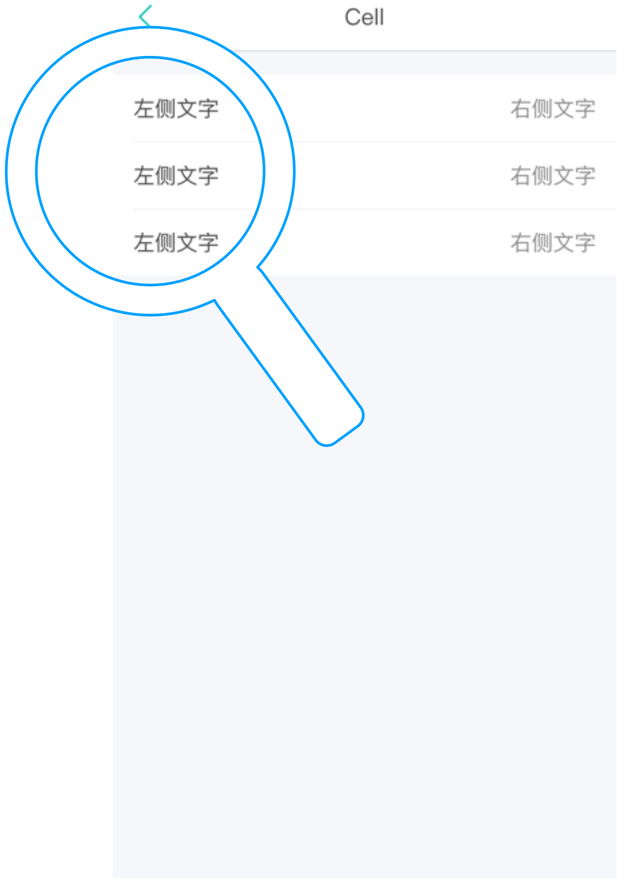
形成一致 UI 风格 — 不一致带来的问题

项目 A

项目 B

形成一致 UI 风格 — 不一致带来的问题

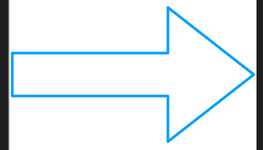




形成一致 UI 风格 — 不一致带来的问题

```vue
<template>
  <div class="cell">
    <div class="cell-left">...</div>
    <div class="cell-right">...</div>
  </div>
</template>

<style lang="less">
.cell {
  padding: 0 0.3rem;
}
.cell + .cell {
  border-top: 1px solid #e5e5e5;
}
</style>
```



```vue
<template>
  <div class="cell">
    <div class="cell-left">...</div>
    <div class="cell-right">...</div>
  </div>
</template>

<style lang="less">
.cell {
  padding-right: 0.3rem;
  margin-left: 0.3rem;
}
.cell + .cell {
  border-top: 1px solid #e5e5e5;
}
</style>
```

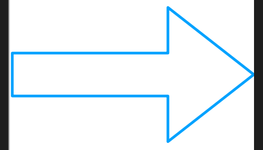
事实并没有这么简单！

如此实现点击高亮区会发生变化！

形成一致 UI 风格 — 不一致带来的问题

```
<template>
  <div class="cell">
    <div class="cell-left">...</div>
    <div class="cell-right">...</div>
  </div>
</template>

<style lang="less">
.cell {
  padding: 0 0.3rem;
}
.cell + .cell {
  border-top: 1px solid #e5e5e5;
}
</style>
```

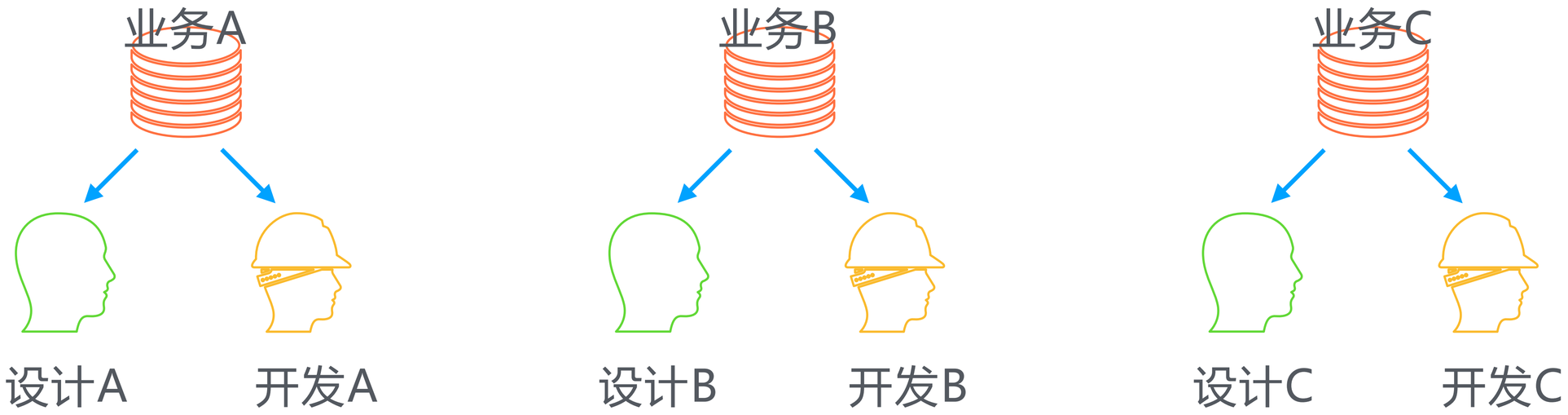


```
<template>
  <div class="cell">
    <div class="cell-inner">
      <div class="cell-left">...</div>
      <div class="cell-right">...</div>
    </div>
  </div>
</template>

<style lang="less">
.cell {
  padding-left: 0.3rem;
}
.cell-inner {
  padding-right: 0.3rem;
}
.cell + .cell .cell-inner {
  border-top: 1px solid #e5e5e5;
}
</style>
```

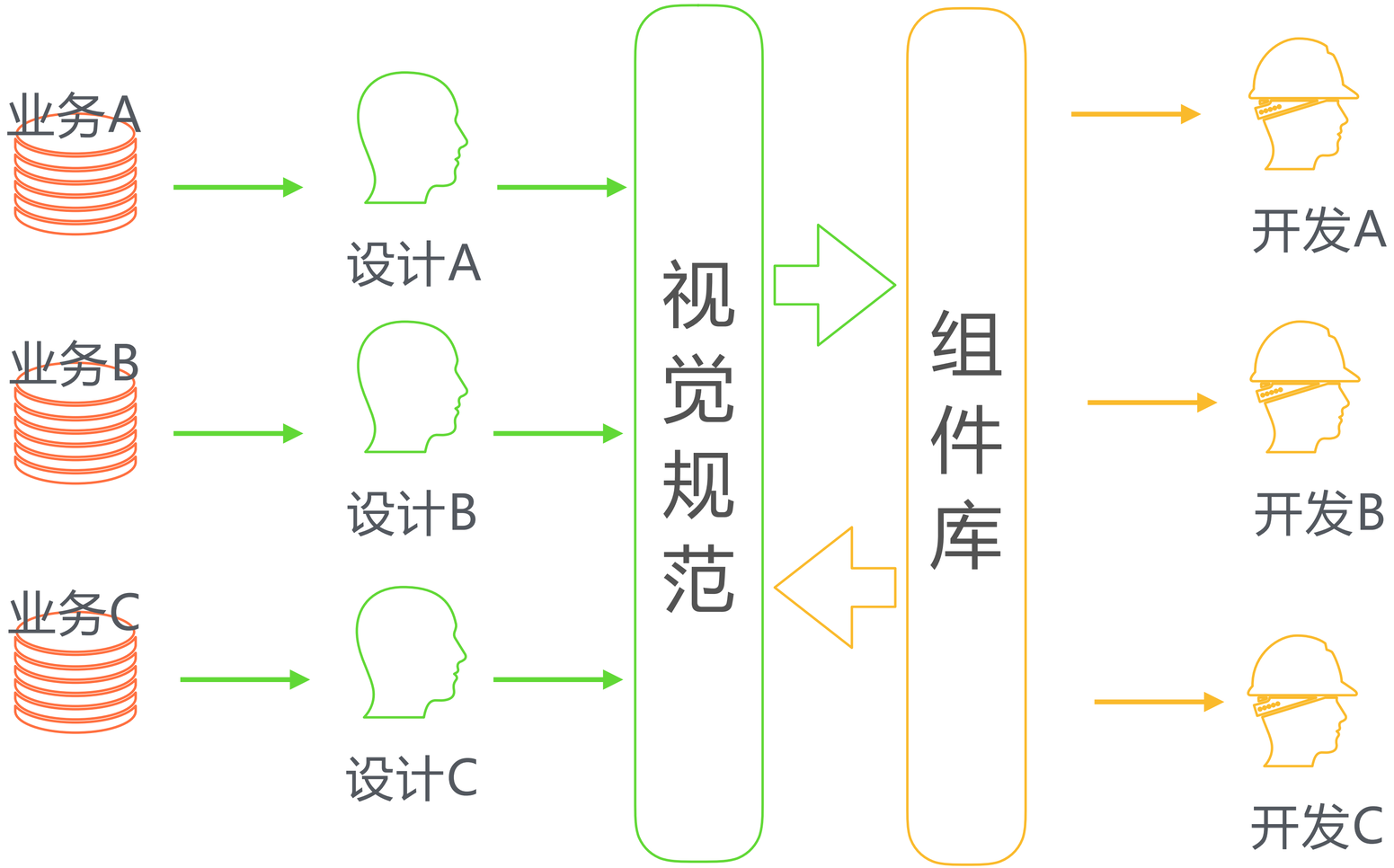
看似小的 UI 细节改动，
实际可能对现有系统会
造成很大影响！

形成一致 UI 风格 — 为什么会不一致

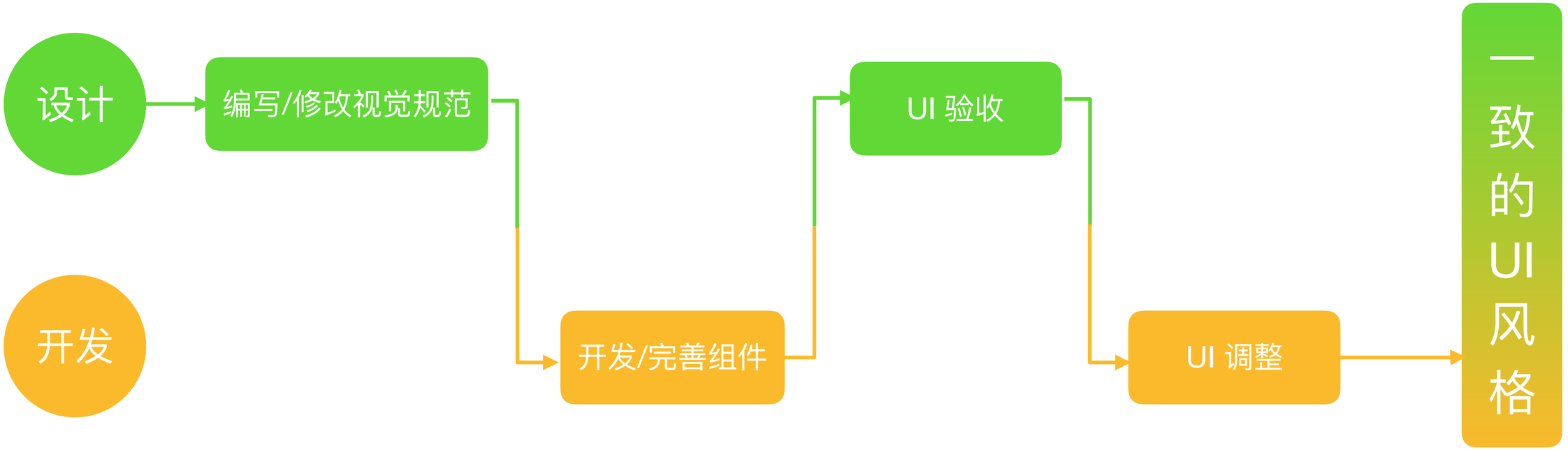


无统一规范　　业务相对独立　　审美差异

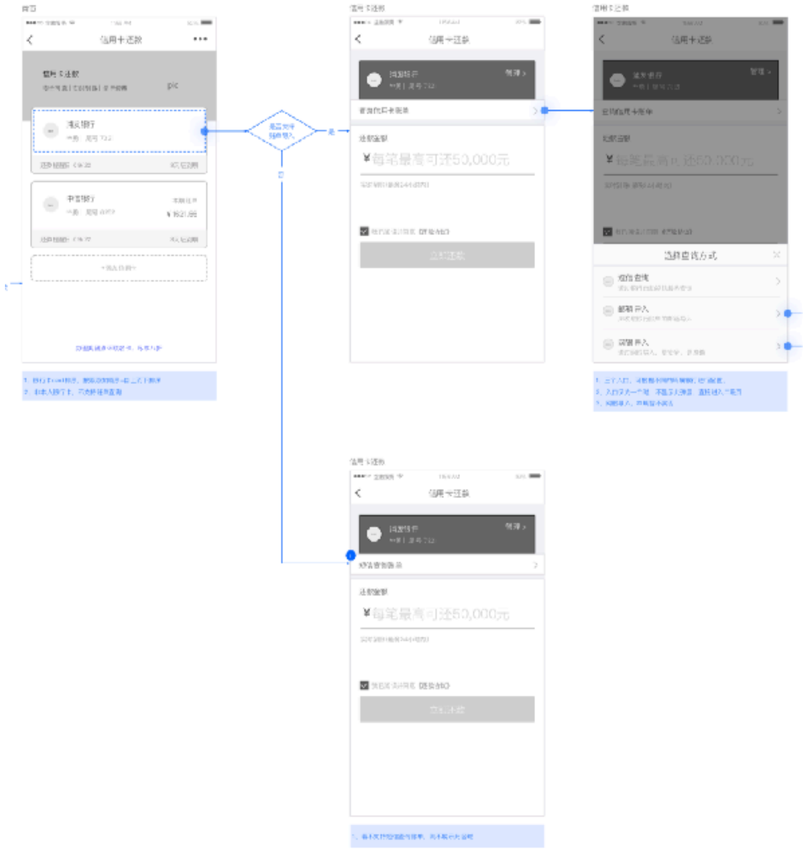
形成一致 UI 风格 — 视觉规范和组件库的一致



形成一致 UI 风格 — 流程规范



形成一致 UI 风格 — 拿到交互即可开发



组件库

发展期

Vix 如何打造可复用性强的组件体系?

可复用性强 — 关于收益的思考

功能点

- 一组有关联的 UI 元素
- 一个完整的交互逻辑
- 一个特定问题的解决

白框可以认为是一个功能点，也可以认为整体是

点击标题后切换内容区可以认为是一个功能点

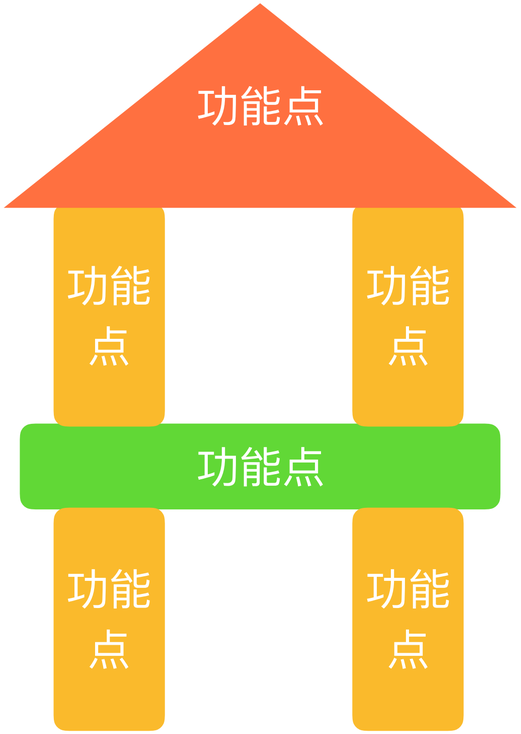
功能点

功能点

功能点

处理点击延迟也可认为是功能点

可复用性强 — 关于收益的思考



单个功能点收益 = 使用 VIX 前开发时间 - 使用 VIX 后开发时间

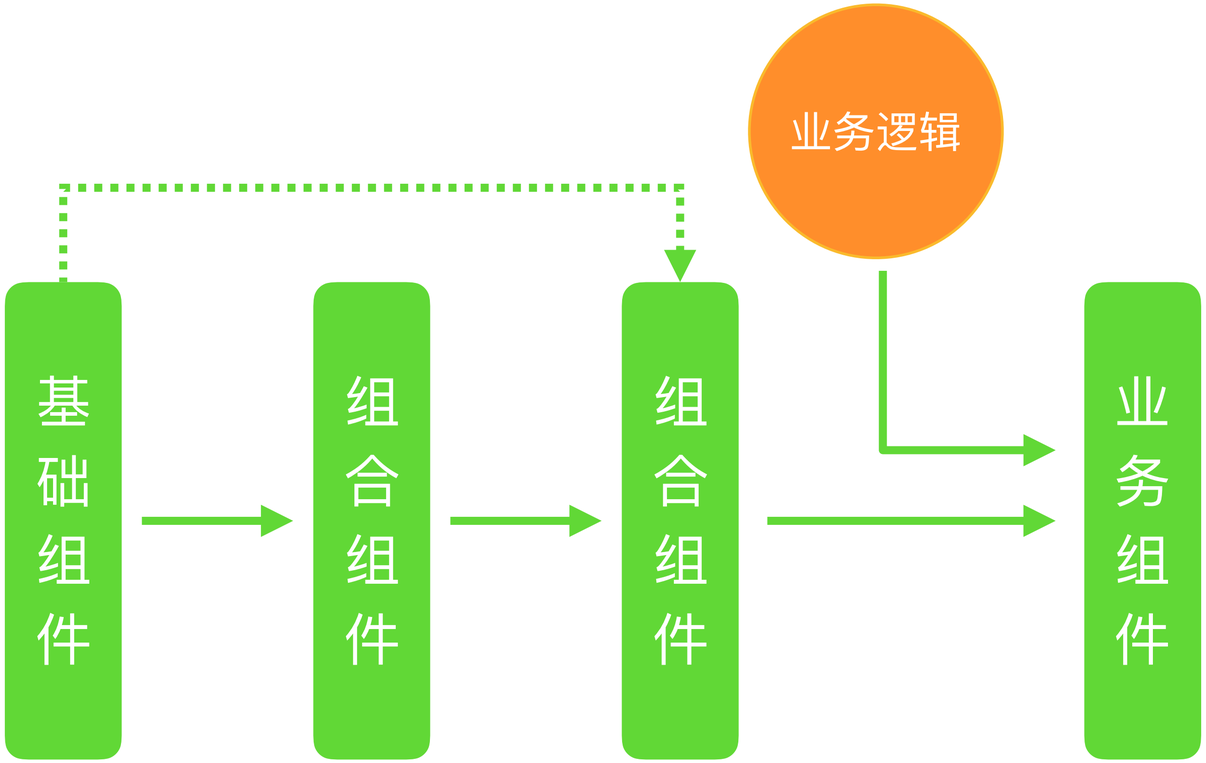
单个项目收益 = $\sum$ 所有功能点收益

1.能覆盖更多功能点

2.单个功能点收益大

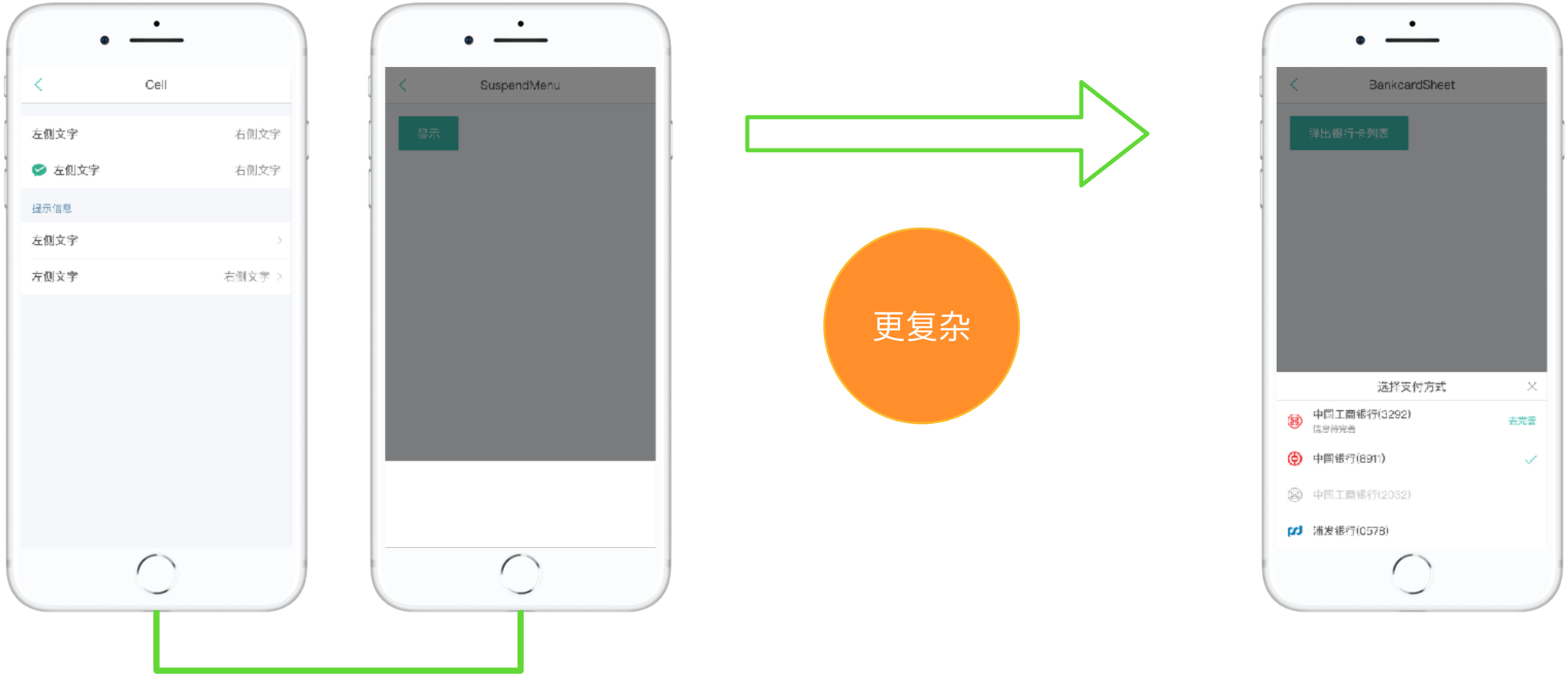
可复用性强

可复用性强 — 关于组件分层



可复用性强 — 如何组件分层

可复用性强 — 如何组件分层



可复用性强 — 开发思想的转换

之前

不能太简单，
解决不了问题

不能太复杂，
失去了通用性

功能需求

现在

层层逼近

1.单个组件职责单一

2.组件对外接口一致

Vix 如何让开发者使用简洁易上手?

简洁易上手 — 成本及优化方法概览

简洁易上手

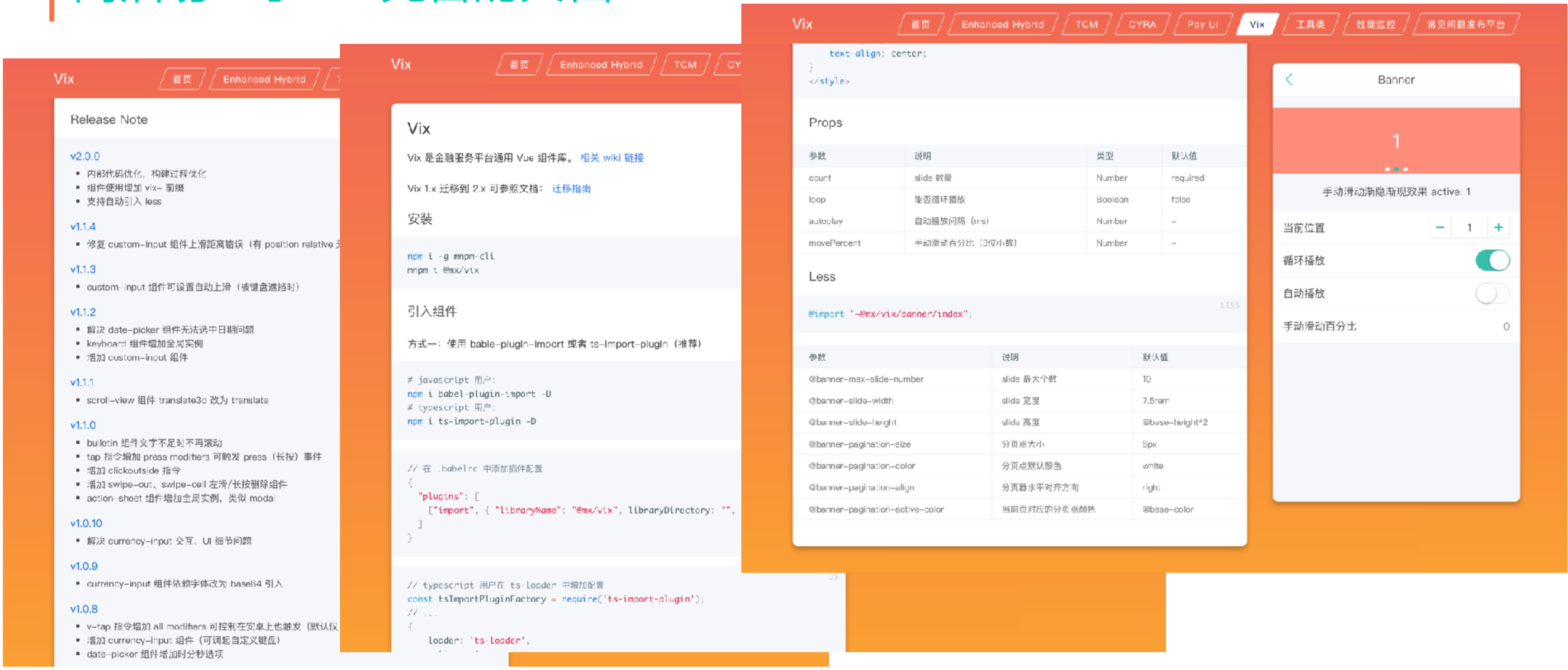
降低理解成本

- 文档完善
- 示例详尽
- 命名规范

降低使用成本

- 引入简单
- 配置简单

简洁易上手 — 完善的文档


Vix
首页
Enhanced Hybrid
TCM
CYRA
Pay UI
Vix
工具类
性能监控
常见问题发布平台
Release Note
v2.0.0
内部代码优化，构建过程优化
组件使用增加 vix- 前缀
支持自动引入 less
v1.1.4
修复 custom-input 组件上滑距离错误（有 position relative
v1.1.3
custom-input 组件可设置自动上滑（被键盘遮挡时）
v1.1.2
解决 date-picker 组件无法选中日期问题
keyboard 组件增加全局实例
增加 custom-input 组件
v1.1.1
scroll-view 组件 translate3d 改为 translate
v1.1.0
bulletin 组件文字不足时不再滚动
tap 指令增加 press modifiers 可触发 press（长按）事件
增加 clickoutside 指令
增加 swipe-out、swipe-cell 左滑/长按删除组件
action-sheet 组件增加全局实例，类似 modal
v1.0.10
解决 currency-input 交互、UI 细节问题
v1.0.9
currency-input 组件依赖字体改为 base64 引入
v1.0.8
v-tap 指令增加 all modifiers 可控制在安卓上也触发（默认仅
增加 currency-input 组件（可调起自定义键盘）
date-picker 组件增加时分秒选项
Vix
Vix 是金融服务平台通用 Vue 组件库。 相关 wiki 链接
Vix 1.x 迁移到 2.x 可参照文档： 迁移指南
安装
npm i -g mnpm-cli
mnpm i @mx/vix
引入组件
方式一：使用 bable-plugin-import 或者 ts-import-plugin（推荐）
javascript 用户:
npm i babel-plugin-import -D
typescript 用户:
npm i ts-import-plugin -D
// 在 .babelrc 中添加插件配置
{
"plugins": [
["import", { "libraryName": "@mx/vix", libraryDirectory: "",
]
}
// typescript 用户在 ts-loader 中增加配置
const tsImportPluginFactory = require('ts-import-plugin');
// ...
{
loader: 'ts-loader',
text-align: center;
}
</style>
Props
参数 说明 类型 默认值
count slide 数量 Number required
loop 是否循环播放 Boolean false
autoplay 自动播放间隔（ms） Number -
movePercent 手动滑动百分比（3位小数） Number -
Less
@import "~@mx/vix/banner/index";
参数 说明 默认值
@banner-max-slide-number slide 最大个数 10
@banner-slide-width slide 宽度 7.5rem
@banner-slide-height slide 高度 @base-height*2
@banner-pagination-size 分页点大小 8px
@banner-pagination-color 分页点默认颜色 white
@banner-pagination-align 分页器水平对齐方向 right
@banner-pagination-active-color 当前页对应的分页点颜色 @base-color
Banner
1
手动滑动渐隐渐现效果 active: 1
当前位置
1
循环播放
自动播放
手动滑动百分比
0

简洁易上手 — 详尽的示例

真实使用场景

最佳代码实践

生动效果体验

简洁易上手 — 规范的命名

统一命名格式

力求精准达意

@action-sheet-z-index	整体 z-index	100
@action-sheet-animation-duration	动画时长	400ms
@action-sheet-animation-timing-function	动画速率	cubic-bezier(0.23, 1, 0.32, 1)
@action-sheet-mask-color	蒙层颜色	rgba(0, 0, 0, .5)

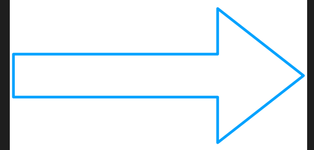
@banner-max-slide-number	slide 最大个数	10
@banner-slide-width	slide 宽度	7.5rem
@banner-slide-height	slide 高度	@base-height*2
@banner-pagination-size	分页点大小	8px
@banner-pagination-color	分页点默认颜色	white

@index-sidebar-z-index	z-index	20
@index-sidebar-text-color	字母表和提示器的字体颜色	@base-grey
@index-sidebar-text-font-size	字母表字体大小	12px
@index-sidebar-indicator-font-size	提示器的字体大小	32px

简洁易上手 — 单行引入

```
<script>
import Comp from 'vix/comp';
export default {
  components: { Comp }
};
</script>

<style>
@import '~vix/base/index.less';
@import '~vix/dep/index.less';
@import '~vix/comp/index.less';
</style>
```



```
<script>
import { Comp } from 'vix';
export default {
  components: { Comp }
};
</script>
```

简洁易上手 — 如何做到单行引入

```
<script>
import { Comp } from 'vix';
export default {
  components: { Comp }
};
</script>
```

webpack 插件

自动转换

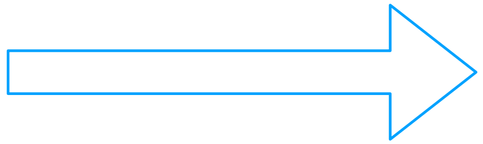
```
<script>
import Comp from 'vix/comp';
import 'vix/comp/style';
export default {
  components: { Comp }
};
</script>
```

Vix 构建时生成

```
import '../base/index.less';
import '../dep/index.less';
import '../comp/index.less';
```

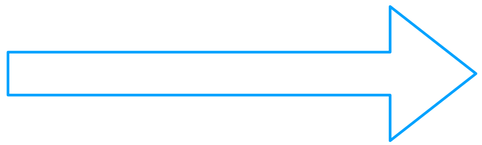
简洁易上手 — 简化参数形式

```
<vix-button type="outline" text="按钮" />
<vix-button status="disabled" text="按钮" />
```



```
<vix-button outline text="按钮" />
<vix-button disabled text="按钮" />
```

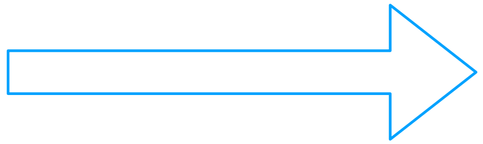
```
<vix-banner v-model="active"
  :count="slide.length"
  :loop="loop"
  :movePercent="movePercent"
  @updateMovePercent="movePercent = $event">
```



```
<vix-banner v-model="active"
  :count="slide.length"
  :loop="loop"
  :movePercent.sync="movePercent">
```

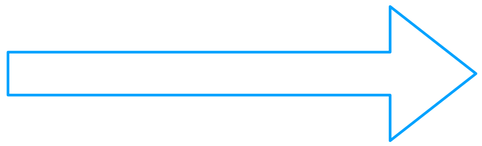
简洁易上手 — 与后端统一数据结构

```
<vix-bill-detail :title="title"
  :logo="logo"
  :status="status"
  :detail="detail"
  :buttons="buttons" />
```



```
<vix-bill-detail :data="data" />
```

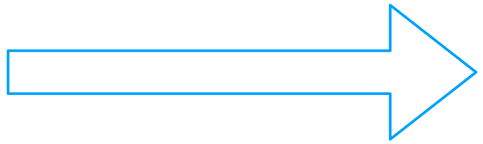
```
<vix-bankcard-sheet :show="show"
  :title="title"
  :notification="notification"
  :bankcardList="bankcardList"
  @hide="onHide">
```



```
<vix-bankcard-sheet :show="show"
  :options="options"
  @hide="onHide">
```

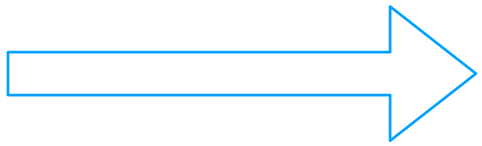
简洁易上手 — 组件内配置合理默认值

```
<vix-bill-detail :title="title"
  :logo="logo"
  :status="status"
  :detail="detail"
  :buttons="buttons" />
```

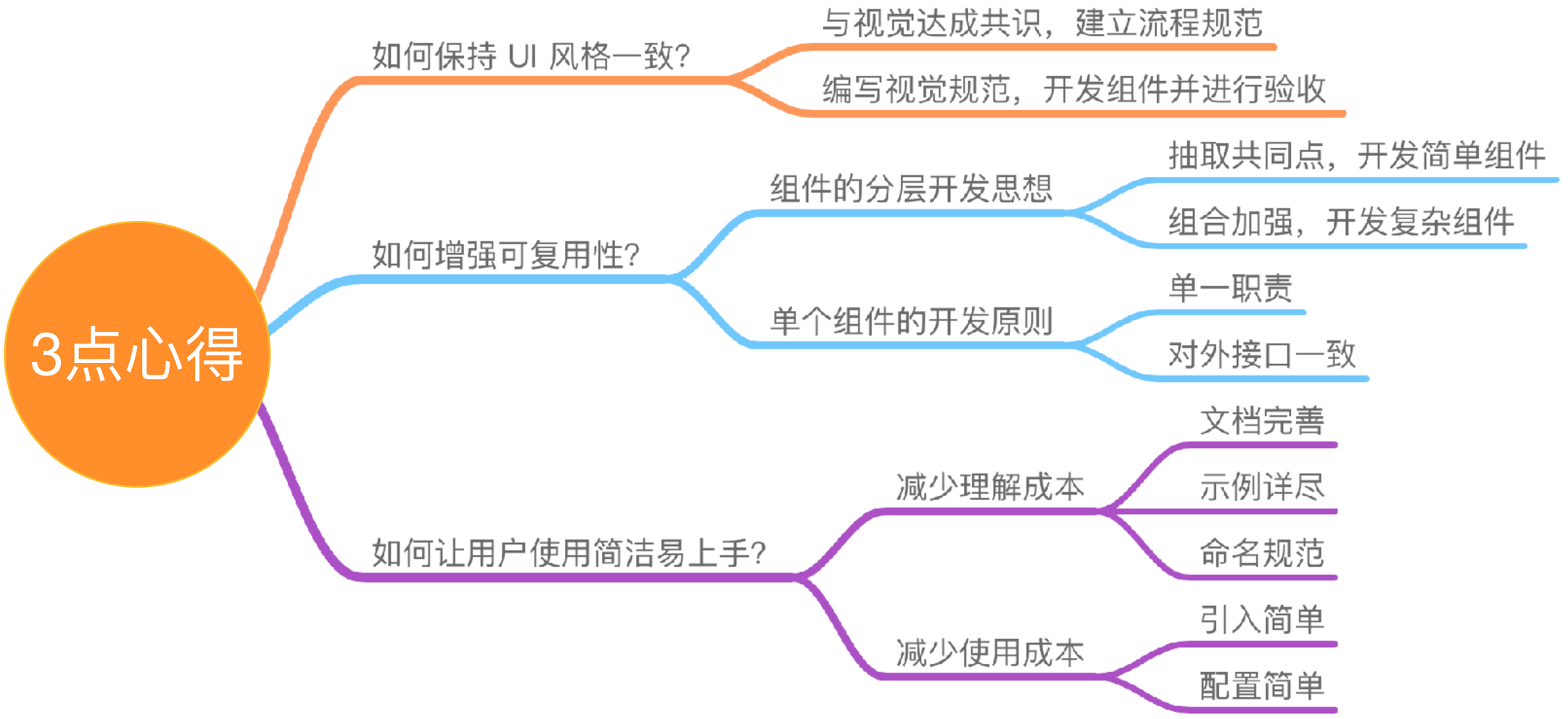


```
<vix-custom-input v-model="amount"
  :placeholder="placeholder" />
```

```
<vix-password :title="title"
  :value="value"
  :info="info"
  @focus="showKeyboard = true" />
```



```
<vix-password @focus="showKeyboard = true" />
```


3点心得
如何保持 UI 风格一致?
与视觉达成共识，建立流程规范
编写视觉规范，开发组件并进行验收
如何增强可复用性?
组件的分层开发思想
抽取共同点，开发简单组件
组合加强，开发复杂组件
单个组件的开发原则
单一职责
对外接口一致
如何让用户使用简洁易上手?
减少理解成本
文档完善
示例详尽
命名规范
减少使用成本
引入简单
配置简单

CODE A BETTER LIFE
一 行 代 码　　亿 万 生 活

Q&A

招聘前端专家、资深前端工程师

简历投放至：chenyulin02@meituan.com

美团