

# 前端模块化估时

2017.10

# 为什么要讨论这个主题

---

- 没有项目管理
- 开发要背锅（加班或为延期负责）

前端模块化估时是一种项目管理工具

# 为什么要按模块估时

---

- 为了少加班
- 为了平稳上线
- 为了提高效率

# 怎么按模块估时（第一步：checklist）

---

- ☐ 需求文档
  - 需求文档还缺什么？
  - 需求文档产品经理是否会认真维护？
- ☐ UI设计图
  - UI设计图还缺什么？
  - 页面上有哪些实体？数据从哪里来？（前端静态：范围是什么；后端返回：后端是否能返回）
  - 如果不是最终版本该怎么处理？
- ☐ 交互图
  - 各种复杂的交互是否都已涉及到？
- ☐ 接口文档
  - 数据是否能和需求文档一一对应？
  - 接口设计是否合理？（什么是合理的接口设计）

## 其他问题

- ☐ 是否需要埋点？
- ☐ 排期是否有硬性要求（活动、承诺）？需求文档、UI设计图、交互图、接口文档什么时候就绪？什么时候提测？什么时候上线？
- ☐ PM开JIRA
- ☐ 拉群

# 怎么按模块估时（第二步：划分模块，预估时间）

模块	预估（天）	实际（天）	完成时间
除折线图之外的所有UI	0.5	0.5	0906am
折线图	0.5		
日期选择器	0.5		
接口联调	1		
codeReview	1		

模块	预估（天）	实际（天）	完成时间
「当前等级」页面静态样式	0.5	0.5	0919pm
「等级攻略」页面静态样式	0.5	0.5	0920am
数据fetch	1	1	0921pm
边界页面和native头部	1	1	0922pm
codeReview	1	1	0923pm（周六）

## 怎么按模块估时（注意事项）

---

- 通过需求文档、UI、交互图、接口文档来估时。如果没有这些输入资料，可以尝试通过交流解决问题，但不能提供准确的估时。
- 组件是否已经存在。如果存在，需要多大的工作量；如果不存在，新开发组件需要多大的工作量。
- 还有哪些难点问题，预估工作量。
- 估时粒度精确到0.5天。

# 再说估时的好处

---

- 为了少加班
- 推动需求评审的完善。
- 为了平稳上线
- 只有记录形成表格，分歧才会明朗，矛盾才会突出。
- 为了提高效率
- 作为沟通工具。如果一些老板过来压缩时间，拿出来给他。
- 作为进度跟踪工具。每天依照这个进度，有安全感。
- 作为数据工具。总结反思可能拿出来。
- 持续分析估时和实际用时的不同，提升估时能力，更加了解自己。

# 参考资料

---

- 《人月神话》
- 《人件》
- 《启示录》
- 《清单革命》



谢谢