

读书笔记: Getting Things Done

阅读 David Allen 的 Getting Things Done (GTD)

1. 介绍

David Allen 的 *Getting Things Done* (GTD) 的核心判断是: 现代人的压力主要不是来自事情太多, 而是来自太多没有被定义清楚的承诺. 这些承诺被 Allen 称为 open loops. 只要一件事还停留在脑子里, 没有被放进可信的外部系统, 大脑就会反复提醒它, 而且经常在完全不能处理它的时候提醒.

所以 GTD 不是先谈优先级, 而是先谈清空心理缓存. 它要做的是把含混的 stuff 变成可以管理的结果和动作: 这是什么? 我想要的完成状态是什么? 下一步可以实际看见的动作是什么? 如果这三个问题没有回答, 一个条目即使被写进待办列表, 也仍然只是换了地方堆放.

2. 工作流

书中最实用的部分是五步工作流.

收集 把所有可能需要处理的东西放进收集篮, 包括纸张, 邮件, 笔记, 想法和物理物品. 收集篮要足够少, 但必须覆盖生活中会产生输入的场景. 关键不是有多少工具, 而是让大脑相信: 它不用自己记.

处理 对每个输入只从上到下处理一次. 先问**这是什么**, 再问**是否需要行动**. 不需要行动的东西只有三类: 丢弃, 稍后再看, 归档参考. 需要行动的东西必须确定项目结果和下一步动作.

组织 结果进入项目清单, 下一步动作进入按场景分组的清单, 固定时间进入日历, 交给别人处理的进入 waiting-for, 以后可能做的进入 someday/maybe. 项目资料和参考资料分开保存, 不要把资料本身当作行动提醒.

回顾 系统只有在被定期回顾时才可信. Weekly Review 的作用是把过去几天散落的输入重新收集, 清空收集篮, 更新项目和行动清单, 重新确认哪些承诺仍然有效.

执行 真正做事时, 选择顺序不是抽象的重要程度先行, 而是先看当前场景, 再看可用时间, 再看精力, 最后才在可做的候选动作里判断优先级.

这套流程的强处在于它把不同性质的活动分开. 收集时不评判, 处理时不拖延, 组织时不重新思考, 回顾时不急着执行. 很多所谓的整理失败, 实际上是把这几件事混在一起做:

一边翻 inbox, 一边决定优先级, 一边试图完成其中几件, 最后既没有清空, 也没有可信的清单.

3. 下一步动作

GTD 里最关心的是下一步动作是什么. 这里的动作必须是物理的, 可见的, 例如写邮件, 打电话, 查资料, 新建文件, 和某人确认. 推进论文或准备旅行都不是下一步动作, 它们只是项目或愿望.

Allen 对行动的分流也很简单:

1. 两分钟内能做完的, 立刻做.
2. 不是自己最适合做的, 委托出去, 并记录 waiting-for.
3. 需要自己稍后做的, 放进对应场景的下一步动作清单.

这个规则的价值不在于两分钟这个数字精确, 而在于它给处理输入设置了一个经济边界: 如果记录, 分类, 追踪一件小事的成本已经超过直接做掉它的成本, 那就直接做掉.

4. 项目计划

书的第三个重要模型是 natural planning. Allen 认为大脑自然规划任何事情时, 通常会经过五层:

1. 明确目的和原则.
2. 想象成功完成后的样子.
3. 发散收集可能相关的想法.
4. 把想法组织成结构, 重点, 顺序或子项目.
5. 决定下一步动作.

很多会议和计划卡住的原因是: 人们常常一开始就问有什么好主意, 但此时目的, 成功标准, 边界条件和材料都还没有被澄清. 结果是大家在不同问题上发散, 却以为自己在讨论同一个项目.

5. 结论

我读下来觉得, GTD 的本质是一套面向知识工作的状态机. 它不保证让人完成更多任务, 而是保证每个输入最终都落到少数几个确定状态里: 丢弃, 参考, 稍后, 项目, 下一步动作, 等待他人, 或者日历. 压力下降不是因为工作消失了, 而是因为未定义的东西减少了.

总结

1. 不把脑子当提醒系统.
2. 看到任何含混事项时, 尽早定义期望结果和下一步物理动作.
3. 每周完整回顾一次所有 open loops.